

AD706866

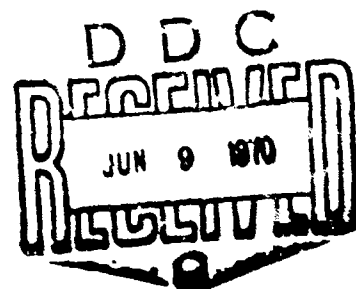
RADC-TR-70-44  
Final Technical Report  
April 1970



ON-LINE RETRIEVAL  
Informatics, Incorporated

Distribution of this document is unlimited.  
It may be released to the Clearinghouse,  
Department of Commerce, for sale to the  
general public.

Rome Air Development Center  
Air Force Systems Command  
Griffiss Air Force Base, New York



208

When US Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded, by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

ACCESSION NO.	
CPTI	WHITE SECTION <input checked="" type="checkbox"/>
DOC	DIFF SECTION <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
REF.	AVAIL. and/or SPECIAL
1	

Do not return this copy. Retain or destroy.

ON-LINE RETRIEVAL

Thomas C. Lowe  
David C. Roberts

Informatics, Incorporated

Distribution of this document is unlimited.  
It may be released to the Clearinghouse,  
Department of Commerce, for sale to the  
general public.

## FOREWORD

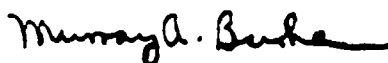
This document is the final report on Contract F30602-69-C-0038, Project 4594, Task 459401, by Informatics, Inc., 4720 Montgomery Lane, Bethesda, Maryland, for Rome Air Development Center, Griffiss Air Force Base, New York. Murray A. Burke, FMIDB, was the RADC Project Engineer. Informatics' number is TR-69-1090-2.

The report summarizes the overall design of an information storage and retrieval system utilizing automatic document characterization and interactive retrieval through man-machine dialogue, and documents in detail the implementation of a portion of that system.

This report has been reviewed by the Information Office (EMLS) and is releasable to the Clearinghouse for Scientific and Technical Information.

This report has been reviewed and is approved.

Approved:



MURRAY A. BURKE  
Project Engineer  
Techniques Development Section

Approved:



A. E. STOLL, Colonel, USAF  
Chief, Intel & Recon Division

FOR THE COMMANDER



IRVING J. GABELMAN  
Chief, Plans Office

## ABSTRACT

This report is concerned with the implementation of an on-line information storage and retrieval system for the Rome Air Development Center. This system is to incorporate techniques of automatic document classification for a large document collection in an interactive environment. Following a review of the system design, the implementation of the system executive is described in detail. Because this executive program also governs communications between the user and the system, it must be a communications package, a training aid, a file building program and an executive program all in one.

## TABLE OF CONTENTS

	<u>Page</u>
SECTION I    INTRODUCTION	I-1
SECTION II    CURRENT STATUS OF THE SYSTEM	II-1
II.1    Historical Background and Introduction	II-1
II.2    Objectives of the RADC On-Line Retrieval System	II-2
II.3    An Overview of the On-Line System	II-4
II.3.1    Indexing and Retrieval Sequences	II-4
II.3.2    Structure of the On-Line System	II-6
II.4    Current Status of the On-Line Retrieval System	II-17
SECTION III    MAN-MACHINE DIALOGUE	III-1
III.1    Functional Description	III-2
III.1.1    The Query Sequence	III-2
III.1.2    The Temporary File	III-10
III.1.3    Query Types	III-11
III.1.4    Levels of Document Information	III-11
III.1.5    Two Retrieval Dialogues	III-12
III.2    Files Accessed by the On-Line System	III-48
III.2.1    Document File--DATA1	III-48
III.2.2    File Words	III-48
III.2.3    Dictionary File	III-48
III.2.4    File CONCEPTS	III-64
III.2.5    File MESSAGES	III-64
III.2.6    File OFFLINE	III-70
III.2.7    Core-Resident Files	III-70
III.3    Flowchart of the Dialogue Processor	III-80

## TABLE OF CONTENTS (cont'd)

	<u>Page</u>
<b>SECTION IV SUBROUTINES CALLED BY THE DIALOGUE PROCESSOR</b>	IV-1
IV.1 Function DOCK	IV-1
IV.1.1 Purpose	IV-1
IV.1.2 Action	IV-1
IV.1.3 Method	IV-5
IV.2 Function LENGTH	IV-5
IV.2.1 Purpose	IV-5
IV.2.2 Action	IV-5
IV.2.3 Method	IV-9
IV.3 Function LOOKUP	IV-9
IV.3.1 Purpose	IV-9
IV.3.2 Action	IV-9
IV.3.3 Method	IV-13
IV.4 Subroutine NUMBER (and Associated Function ZORCH)	IV-14
IV.4.1 Purpose	IV-14
IV.4.2 Action	IV-14
IV.4.3 Method	IV-18
IV.5 Subroutine CUT	IV-19
IV.5.1 Purpose	IV-19
IV.5.2 Action	IV-19
IV.5.3 Method	IV-19
IV.6 Subroutine PLUCK	IV-19
IV.6.1 Purpose	IV-19
IV.6.2 Action	IV-24
IV.6.3 Method	IV-25
IV.7 Function PUT	IV-26
IV.7.1 Purpose	IV-26
IV.7.2 Action	IV-26

## TABLE OF CONTENTS (cont'd)

	<u>Page</u>
SECTION IV SUBROUTINES CALLED BY THE DIALOGUE PROCESSOR (cont'd)	
IV. 8 Subroutine STEM	IV-26
IV. 8.1 Purpose	IV-26
IV. 8.2 Action	IV-30
IV. 8.3 Method	IV-30
IV. 8.4 Example	IV-31
IV. 9 Subroutine WHERE	IV-31
IV. 9.1 Purpose and Action	IV-31
IV. 9.2 Method	IV-36
IV. 9.3 Deactivation	IV-36
IV. 10 Function YESNO(I)	IV-36
SECTION V FURTHER WORK	V-1
SECTION VI LISTINGS	VI-1
SECTION VII REFERENCES	VII-1

## ILLUSTRATIONS

	Page
<b>SECTION II    CURRENT STATUS OF THE SYSTEM</b>	
II-1    On-Line System Structure	II-7
II-2    On-Line File Structures	II-9
II-3    Example of System Operation	II-13
II-4    Simple Query	II-15
 <b>SECTION III    MAN-MACHINE DIALOGUE</b>	
III-1(a) Dialogue with Inexperienced User	III-13
III-1(b) Dialogue with Experienced User	III-30
III-2(a) Files Accessed by the On-Line System: GECOS III Files	III-49
III-2(b) Files Accessed by the On-Line System: Core-Resident Files	III-50
III-3    DATA1	III-51
III-4    WORDS Listing	III-53
III-5    Stem Dictionary	III-56
III-6    DICTIONARY GENERATOR	III-60
III-7    Dictionary	III-62
III-8    Characteristics of Test Document Collection	III-65
III-9    CONCEPTS File	III-66
III-10   Program CONGRA	III-68
III-11   MESSAGES File	III-71
III-12(a) Families of Arrays	III-77
III-12(b) Program Variables and Arrays	III-78
III-12(c) Common Storage	III-79
III-13   DIALOGUE PROCESSOR	III-81
 <b>SECTION IV    SUBROUTINES CALLED BY THE DIALOGUE                  PROCESSOR</b>	
IV-1    Directory of Programs and Subprograms	IV-2
IV-2    Function DOCK	IV-3
IV-3    Function LENGTH	IV-6
IV-4    Function LOOKUP	IV-10

## ILLUSTRATIONS (cont'd)

		Page
SECTION IV	SUBROUTINES CALLED BY THE DIALOGUE PROCESSOR	
	IV-5 Subroutine NUMBER	IV-15
	IV-6 Subroutine OUT	IV-20
	IV-7 Subroutine PLUCK	IV-21
	IV-8 Demonstration of PLUCK	IV-23
	IV-9 Subroutine STEM	IV-27
	IV-10(a) File CHICKEN	IV-32
	IV-10(b) Results of STEM	IV-33
	IV-11 WHERE	IV-34
	IV-12 YESNO	IV-37
 SECTION VI	 LISTINGS	
	VI-1 Function DOCK	VI-2
	VI-2 Function LENGTH	IV-4
	VI-3 Function LOOKUP	VI-7
	VI-4 Subroutine NUMBER, Function ZORCH	VI-10
	VI-5 Subroutine OUT	VI-13
	VI-6 Subroutine PLUCK	VI-14
	VI-7 Function PUT	VI-16
	VI-8 Subroutine STEM	VI-17
	VI-9 Subroutine WHERE	VI-20
	VI-10 Function YESNO	VI-21
	VI-11 Program DIALOGUE	VI-22
	VI-12 Program CONGRA	VI-41
	VI-13 Program DICGEN	VI-43

## SECTION I

### INTRODUCTION

This document is the final report on the development of an on-line information storage and retrieval system for the Rome Air Development Center, Air Force Systems Command, Griffiss Air Force Base, New York. Under this contract, an on-line storage and retrieval system, called for brevity in this report the On-Line System, has been designed, and its executive, called in this report the dialogue processor, has been programmed. The dialogue processor has been provided with routines to simulate the rest of the On-Line System, so that to the user, the entire System appears to be implemented.

An overview of the design of the On-Line System is first presented, along with a summary of the present status of the dialogue processor and its supporting programs. The operation of the dialogue processor is then described in detail, and two examples of actual user dialogues with the dialogue processor are presented to illustrate the discussion. Construction techniques for the files it accesses are presented. A description of the subprograms that make up the dialogue processor and a discussion of useful areas for further work conclude the report.

## SECTION II

### CURRENT STATUS OF THE SYSTEM

This section presents the current status of the On-Line System. An overview of the design of the System is presented, including a discussion of the user and System actions that take place during a query sequence. The section concludes by discussing in detail the present status of both the System design and the dialogue processor.

#### II.1 HISTORICAL BACKGROUND AND INTRODUCTION

The computer is a potentially powerful tool for browsing through vast quantities of information. The speed and storage capacity of modern computer systems promise to make the resources of a library available without the huge investment of time required to establish, maintain, and use the manual searching aids usually associated with a library.

Much work has been done in the development of on-line systems, and additional background on other systems is provided in other papers (1, 2, 3, 4, 6, 7, 9). However, all of these systems retrieve by means of simple coordinate indexing and various embellishments on it. Only one type of ranking exists in such systems--the identification of relative relevance of retrieved documents. It is obtained from a tally of the number of elements of "or" clauses retrieving each document. This form of ranking is crude in that it does not give a very sensitive measure of relevance.

Most such systems rely on manual indexing of documents and retrieve on descriptors. Some allow the use of an on-line thesaurus; some also allow retrieval on title or author's name. A few allow searching on partial words and word phrases. Thus, most present on-line systems rely on manual indexing (except for title and author information) and perform retrieval based on logical connectives. Except for provision of thesauri, little is done to assist the user's synonym problems.

The size of collections presently being encountered, their growth rate, the scarcity of competent indexing personnel, and the cost of manual indexing have promoted the search for automated methods of document classification for retrieval (8). This includes not only the actual indexing of documents, but the development of thesauri. Naturally, the development of new retrieval techniques is intimately linked with work in classification methods.

## II. 2 OBJECTIVES OF THE RADC ON-LINE RETRIEVAL SYSTEM

The RADC On-Line Retrieval System is an attempt to overcome many of the difficulties that have been associated with information storage and retrieval systems through the use of a new approach to the problem of on-line retrieval--the concept vector technique. This System is to enable classification of documents to be performed automatically; the correlation that can be performed on documents indexed by concept vectors should be far superior to that which can be achieved with coordinate indexing. The use of the concept vector technique will also permit retrieval based on similarity to any specified document in the collection.

Concept vector indexing is performed in batch mode for use within the framework of a fully interactive on-line system. The on-line nature of the retrieval System operation imposes fundamental constraints on the entire System design, if the result is to be useful.

Batch system queries are frequently written by a system "expert" who interprets the information requests submitted by users. However, in interactive systems, the user himself formulates queries and operates the system. Therefore, for successful operation, the on-line dialogue must be easy and natural to use.

A user must be able to concentrate on the problems of retrieving information, and not be required to second-guess the designers of the System. Users with differing levels of familiarity are to be expected. The inexperienced user must be led through the System step-by-step, whereas the

experienced user should be able to exercise a great deal of flexibility in employing the System. The messages from the experienced user to the System would be expected to be terse, whereas those from the neophyte would be more verbose and tutorial.

Consider the additional power of an on-line system if the user is given the ability to locate documents that are in some way like a known document. This can be illustrated by considering the problem of finding documents in the stacks of a conventional library. Suppose that one could only request documents by their classification numbers and that one were not allowed to enter the stacks. Depending on the user's familiarity with the classification system and with the document collection, he might or might not be able to retrieve all of the documents relevant to his needs.

Now, if the user can enter the stacks of the library and browse about, his chances of finding useful documents are increased. They are likely to be physically near the documents specified initially (and, of course, may include those documents). The hierarchical classification scheme of the library has been mapped into one-dimensional space: the ordering of the books in the stacks. An on-line system can be built in such a way that the user is free from the constraints of a space of limited dimensionality and can search for documents "like" a given document. This is known as document-document searching and is analogous to browsing in a library where every intellectual area (or "concept") corresponds to a different dimension.

A further novelty of the RADC On-Line System concerns the size of the data base to be indexed and accessed; it will eventually contain more than 100 million characters of text. The size of the data base presents particular problems in the design of the off-line programs that perform indexing. The indexing processes must be designed to avoid rapid growth of core requirements as the data base size increases. For example, where as storage of a similarity matrix for 100 documents requires 10,000 similarity coefficients to be computed and stored, the same matrix for 1,000

documents would have 1,000,000 elements. Thus, many processes that are useful for small document collections simply cannot be used with a large data base.

## II.3 AN OVERVIEW OF THE ON-LINE SYSTEM

This subsection presents a highlight of the most significant design features of the RADC On-Line Retrieval System. It is noted that the present System does not include all the designed features.

### II.3.1 Indexing and Retrieval Sequences

Although the On-Line System operates on concept vectors, it must use a thesaurus. The thesaurus contains word stems rather than words, and is automatically developed from the document file. First, common words (e.g. a, an, the) are removed and stem analysis is employed in order to select the distinct noncommon stems occurring in the document collection. This large group of stems is reduced to a smaller collection of so-called content stems, which constitutes the thesaurus. This selection of the content stems from the collection of raw stems is to be performed by the statistical filtering program, which selects those word stems most promising for the characterization of documents. It does this by analysis of both the stem rank-frequency distribution and the variation of that distribution over the document collection.

With every document is associated its concept vector. This vector consists of concept-weight pairs. A concept vector can be formed from any body of text; therefore, in order to perform a retrieval query, it is only necessary to derive the concept vector for the query and correlate it with concept vectors for the documents in the collection. Those documents with vectors producing the highest correlation are then retrieved.

The concepts themselves could, of course, be word stems. However, this would not allow the System to account for the use of words that are similar in meaning, and would introduce one of the worst drawbacks of simple coordinate indexing--the need for the user of the System to consult a thesaurus of "use" and "used for" terms. Instead of this stem-per-concept approach, the System is to cluster stems into about 1500 groups. Each group contains stems of similar semantic value, and each group corresponds to a concept. The clustering is to be performed on a basis of statistical stem co-occurrence analysis.

An ordinary retrieval on the basis of a text query is performed in the following manner. First, the user's request is processed by the routines which reject common words and perform stem analysis, reducing the query to a sequence of stems. Each concept stem is then mapped by a dictionary processor into one or more clusters. Since each cluster is associated with a concept, this process produces the concept vector corresponding to the query. This vector can be correlated against the concept vectors for the document collection in order to perform the retrieval. In order to avoid comparison with all the concept vectors for a large collection, say, 40,000 documents, the document concept vectors themselves are clustered about centroids. This materially reduces the search time.

As mentioned in the last section, document-document correlation can also be performed by the On-Line System. This form of searching simply employs the concept vector of a known document in order to retrieve similar documents. (It is also possible for the user to construct and modify query concept vectors directly, working only with numeric concept codes and weights.)

During the retrieval process, the user can be expected to try a number of queries. Some will retrieve desirable documents, and some will not. The user is given the capability to build a file of documents, retaining those which he finds desirable.

### II. 3. 2      Structure of the On-Line System

Figure II-1 shows the overall structure of the On-Line System. Files are represented by symbols with rounded sides; rectangles represent programs. An arrow from A to B indicates that A calls B, if A and B are programs. If B is a file and A is a program, the arrow indicates that A writes on B; if A is a file and B is a program, then B reads from A.

The dialogue program keeps track of the status of the present query sequence by maintaining the query sequence status file. Because this file contains the information needed to direct the operation of the other programs in the On-Line System, the dialogue program performs the executive function and is resident in core at all times while the on-line system is in operation. For this reason, the core requirements of the dialogue program must be minimized; therefore, the only file that DIALOGUE will keep in core is the query sequence status file which will contain the current query words, stems, concept numbers, weights, and various flags that specify the status of the query.

The four program modules that are loaded into core by the dialogue program are shown in Figure II-1 as the four blocks immediately below the dialogue program. Each of the program modules will be loaded with the subprograms that it calls. With one exception, CHOOSE, only one of the four program modules will be resident in core at once.

II. 3. 2. 1      Files.    The entities shown as files in Figure II-1 are not necessarily distinct files that will be stored on auxiliary storage devices; rather, every sizable data structure is identified here as a file so that an explicit decision concerning its residence can be made.

The four files shown on the left margin of Figure II-1 are arranged hierarchically in order of increasing minimum access time requirements. Exactly which file is resident on what type of auxiliary storage device is a decision to be based upon both the amount of auxiliary storage available

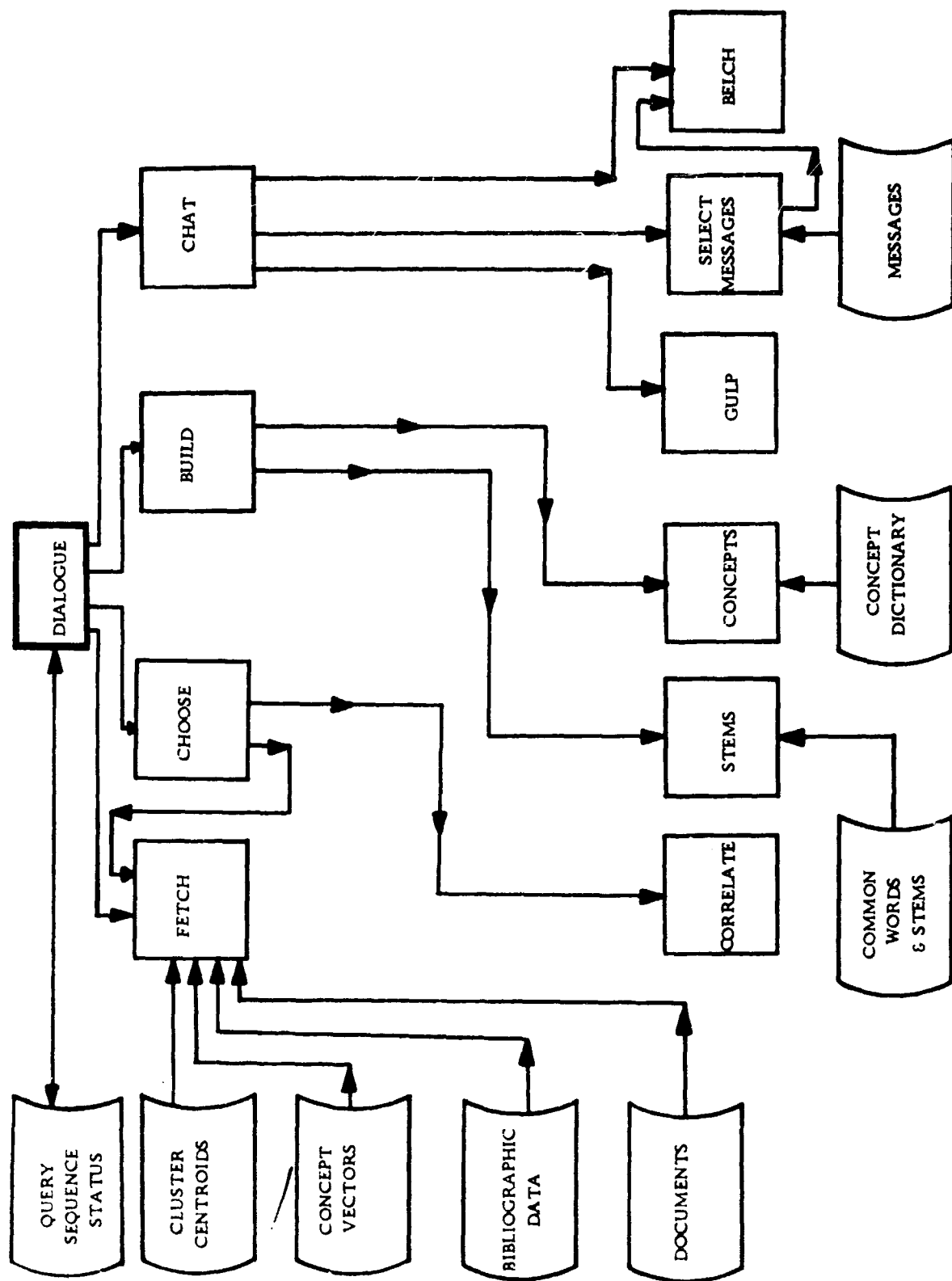


Figure II-1 On-Line System Structure

and response time requirements. For the System to interact conversationally with the user, at least the cluster centroids and concept vectors must be stored on a high-speed, direct-access storage device. The document file can be allocated to tape or disk storage.

The file structure has been designed to accommodate the widest possible variation in data base characteristics. The main contributor to this flexibility is the use of variable-length records in every file. This not only removes the need for some arbitrary limit on the size of each type record, it also greatly increases the efficiency with which the available disk storage space is used, because every record will occupy only the amount of space it requires.

The on-line files that will be accessed by program module FETCH are:

1. Documents
2. Bibliographic data
3. Concept vectors
4. Centroids.

Before the on-line system can be used, these data must be loaded into four distinct GECOS III permanent files by SHOVEL. Four separate files are used in order to permit all the records that are associated with a given document in the data base to be obtained by using only the accession number of the generating document. Because of this, no separate directory will be necessary, and cross-referencing from a concept vector to a bibliographic record to the document itself can be performed without intermediate accesses to a directory.

Figure II-2 illustrates the organization of the on-line files. The solid arrows represent an explicit "pointing" relationship; the dashed arrows represent an implicit "pointing" relationship that arises because

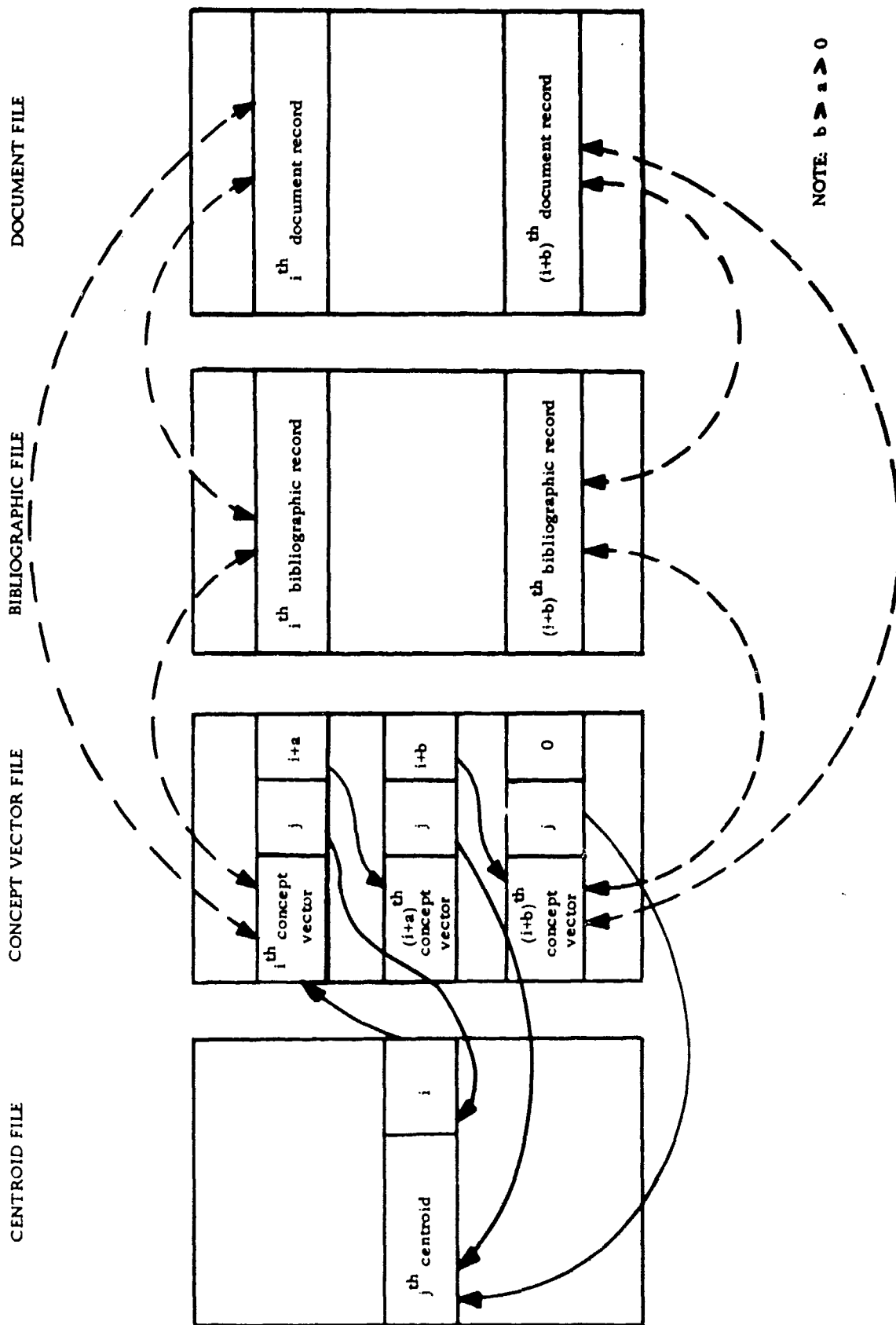


Figure II-2 On-Line File Structures

the concept vector, bibliographic, and document files are all ordered by accession number. Thus, the arrows indicate all the possible methods of cross-referencing the various files.

In order that the files can be organized efficiently by accession number, it is required that the accession numbers be a compact set of positive integers starting with one. If the data base is supplied without these integral accession numbers, it is a simple matter to number all the documents.

The use of a distinct file for each class of data also permits the selective loading of the various files. The On-Line System might be used for experiments that would not access all of the files. In this case, the selective loading of the on-line files, by reducing disk usage, will increase operational economy beyond that which might otherwise be associated with experimentation with the full On-Line System.

**2.3.2.2 Program Modules.** This subsection introduces each program module and gives a brief description of its function.

The primary function of DIALOGUE is to keep track of the user's status and direct user-system interaction. Therefore, DIALOGUE maintains the information needed to direct the sequence of operation of the other program modules and also serves as the executive of the On-Line Retrieval System.

Program module FETCH performs all accesses to the on-line data base. Given a record number and a file designation, FETCH returns the record and size of the record. FETCH obtains only one record at a time; to obtain all the records in a file, FETCH must be called repeatedly.

FETCH will be loaded by itself or together with CHOOSE. FETCH will be loaded by itself when a file access is being performed that does not require selection of concept vectors based on their correlation with some query vector, such as when scanning of the bibliographic data or document file is taking place.

Program module CHOOSE, given a query vector by DIALOGUE, returns to DIALOGUE the accession numbers of the documents whose concept vectors have the highest correlation coefficients with the query vector. In order to do this, CHOOSE calls FETCH to obtain the centroids of all clusters, and then calls CORRELATE to determine which clusters to scan. When this is complete, FETCH is called to obtain the selected clusters, and the concept vectors in these clusters are similarly processed by CORRELATE. CHOOSE then returns to DIALOGUE the accession numbers of the documents whose concept vectors correlate most highly with the query.

Program module BUILD operates on a list of words and produces a concept vector. It does this by first performing stem analysis by calling STEMS, then mapping the stems into concepts by calling CONCEPTS. Program STEMS includes within it the list of common words and the list of stems to be removed; program CONCEPTS includes within it the dictionary of content stems and the concept numbers and weights into which each is mapped.

Each word in a query can fall into one of three categories. It may be a common word that is deleted by STEMS, a word that generates a noncontent stem, and, therefore, is not mapped into a concept, or a word that generates a content stem, and, therefore, is mapped into one or more concepts. BUILD will recognize and differentiate between these three cases and report this information to DIALOGUE along with the generated concept vector and stems.

BUILD will be called to process a query before calling CHOOSE. When document-document correlation is being performed, BUILD will not be used, since the query vector in that case will be obtained by using FETCH to access the concept vector file.

Program module CHAT communicates with the user. Standard On-Line System messages are sent to the user by calling SELECT. Given a message number, SELECT accesses the file of messages, selects one, and calls BELCH to transmit the message. BELCH transmits one line to the remote terminal; GULP reads a line from the terminal.

When documents are being printed at the remote terminal, SELECT will not be used. DIALOGUE will obtain the data to be sent by calling FETCH, and then call BELCH to transmit. Data obtained from other program modules, such as BUILD, will also be transmitted without a call to SELECT.

II.3.2.3 Examples of System Operation. Figure II-3 illustrates the roles played by the various program modules by showing the sequence of events that might take place during the processing of a query. This example shows only the gross features of query processing and document document correlation; a sophisticated user would cause a much more complex process to occur.

During operation of the system, DIALOGUE performs a function in addition to those shown explicitly in the flowchart; it directs the loading of the other program modules.

The user begins the sequence by entering a query which is read by CHAT. BUILD is then loaded and performs stem and concept analysis, producing a concept vector if the query contains any words that generate content stems. DIALOGUE stores this concept vector as the query vector, and loads CHOOSE and FETCH together. By calling FETCH and CORRELATE,

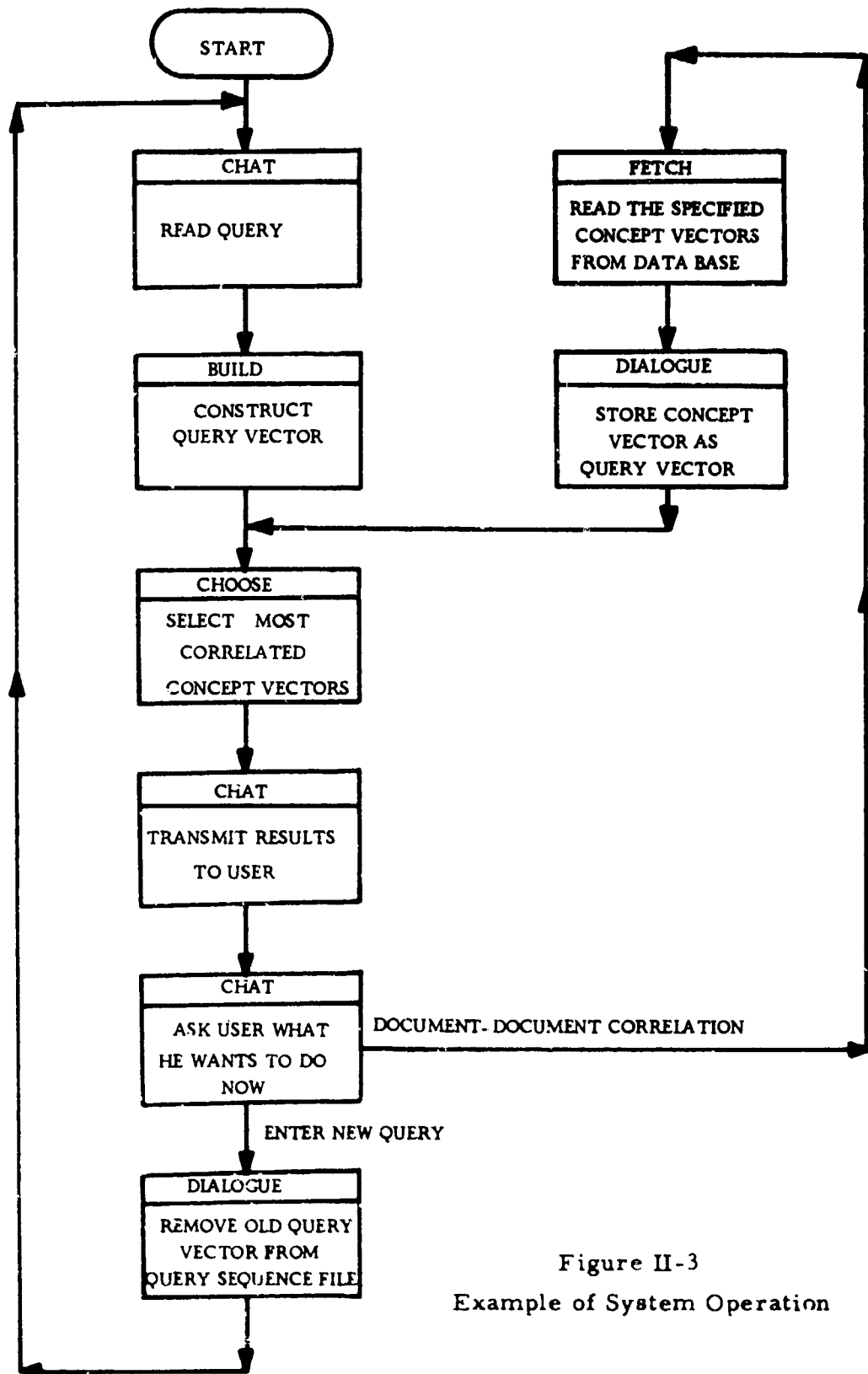


Figure II-3  
Example of System Operation

CHOOSE determines the accession numbers of the documents whose concept vectors correlate most highly with the query vector. This list is passed to DIALOGUE.

When DIALOGUE has received the query results, it loads CHAT to transmit the results to the user. At this point, the user might elect to enter a new query, in which case DIALOGUE clears QUERY SEQUENCE STATUS, or he might elect document-document correlation. He also has several other options which are not shown in this example.

Document-document correlation is performed by using FETCH to obtain the concept vectors that are to be used as query vectors, and then calling CHOOSE in the same fashion as when processing a user-generated plain text query.

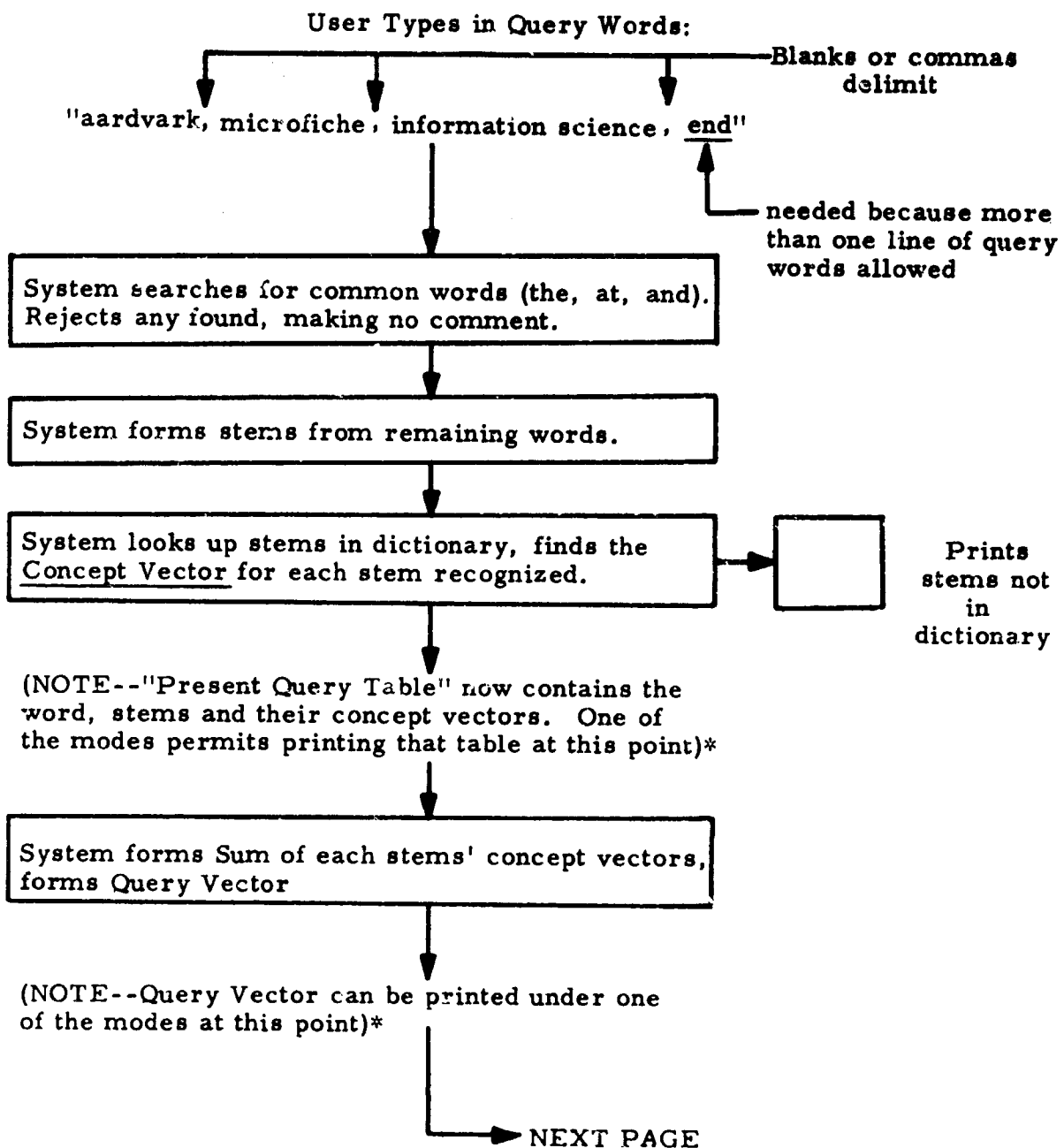
Figure II-4 shows the sequence of events that might transpire during the processing of a simple query. This figure emphasizes the System and user actions, whereas Figure II-3 identifies the specific program modules that perform each action.

The sequence begins when the user enters his initial query.\* The System first identifies and removes any common words from the query, without comment. The System then performs stem analysis on the remaining words, mapping each word into a sort of "canonical form" for its morpheme.

Each stem is then looked up in the stem-concept dictionary, and the concept codes and weights thus obtained are added to form the query concept vector. The stems from the query that were not found in the dictionary are printed, so that the user can decide whether he wishes to perform retrieval with his query as it stands, or add more words.

---

\* An experienced user might start differently.



\* "Normal Mode" Suppresses this--average user probably does not want it.

Figure II-4 SIMPLE QUERY (cont'd)

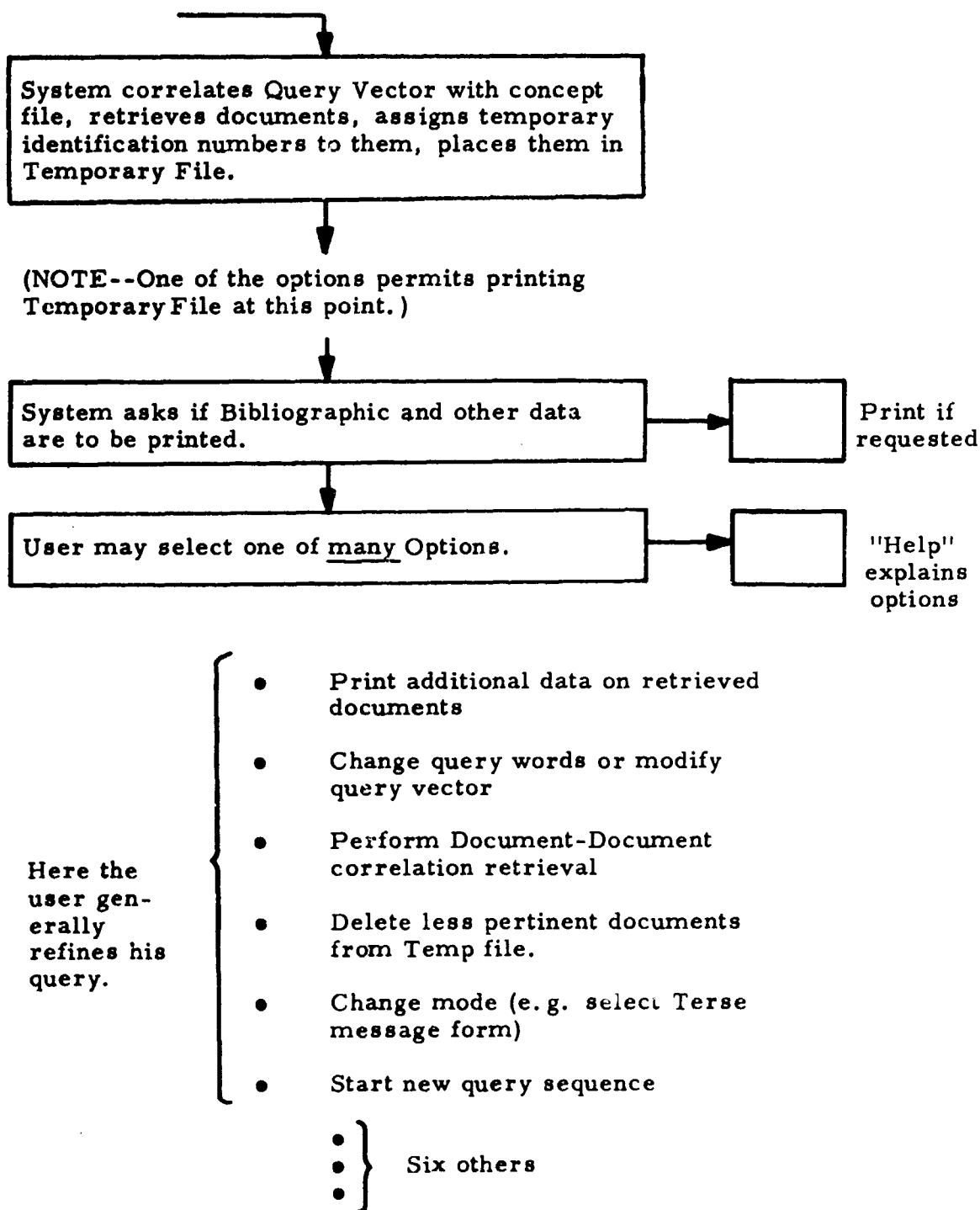


Figure II-4 SIMPLE QUERY (concluded)

Once the user decides to retrieve with his query, the System correlates the query vector with the concept vector for each document in the collection. The System places a ranked list of the highest-correlated documents in the collection in a temporary file, for access by the user.

The System then informs the user that the retrieval is complete. He may then print the contents of the temporary file, which will tell him not only which documents were retrieved, but also their correlations with the query concept vector and their rank by correlation. Also, a temporary identification number is assigned to each document, so that the user may refer to documents without typing in a lengthy accession number.

Once the temporary file of results from a retrieval has been formed, the user has a number of ways in which he can use the results. He can print bibliographic data for all, some, or one of the retrieved documents; he can print any of the documents themselves, or he can use the retrieved documents to find other similar documents in the collection.

#### II. 4 CURRENT STATUS OF THE ON-LINE RETRIEVAL SYSTEM

The System design, including the design of the off-line indexing programs, is complete and is presented in detail in the Interim Report (5). Although the present System design does not exhaust the potential capabilities of automatic indexing, and further study can still produce significant results, nevertheless the implementation of the full On-Line System is clearly feasible for a document collection containing over 100 million characters of text.

The heart of the On-Line System, the dialogue processor, has been programmed, in GECOS III Time-Sharing FORTRAN, and has been provided with additional routines to perform other System functions in order

to permit experimentation with the user-System interface. The dialogue processor is thus completely operational, and successful query sequences have been accomplished, using a test collection of 50 documents provided by RADC.

In its present configuration, all features of the user-System interface are present. Therefore, to the user, the presently operating portion of the System appears to be the entire System. Several detail refinements of the user-System interface have been made as a result of experiments conducted with the dialogue processor. This prototype will still be useful as an easily-modified test bed even when the entire System has been implemented.

## SECTION III

### MAN-MACHINE DIALOGUE

The heart of the On-Line System is the software module which governs communications between the user and the System. This module--the dialogue processor--also performs the executive function of the On-Line System. It calls on the routines which perform stem analysis, retrieval, ranking, dictionary lookup and all the other functions. It solicits queries and commands from the remote user, causes search queries to be executed, and reports and stores the results and generally leads the user through the array of tools available to him in his searching of the data base. The dialogue processor is, therefore, a communications package, a training aid, a file building program and an executive program all in one.

The dialogue processor has been programmed, and is fully operational. Although the other program modules that are called by the dialogue processor (see section II.3.2) have not been programmed, the dialogue processor has been provided with supporting routines that simulate the operation of these modules. In this manner, to the user, the entire On-Line System appears to be implemented. Thus, the two sample dialogues included in this section are essentially identical to dialogues that will be conducted with the full On-Line System.

This section presents a functional description of the dialogue processor in subsection III.1. The discussion is then illustrated by two actual retrieval dialogues, as might be conducted by two users of differing experience levels. These two examples show how the novice and the experienced user might use the same query to obtain information with the System.

Crucial to the operation of the dialogue processor are the various files that it accesses; in subsection III.2, the format and method of construction of each file is discussed.

Section III.3 contains a flowchart of The Dialogue Processor, in Figure III-13.

### III.1

#### FUNCTIONAL DESCRIPTION

The dialogue processor is designed, insofar as its functional characteristics appear to the user, with the overriding concept that different users of differing ability, needs, familiarity and goals will at various times attempt to use the System. In order for these attempts to succeed, the System must be geared to the user. The experienced user will not tolerate the delays incurred as lengthy tutorial messages are printed at the Teletype terminal; the inexperienced user will flounder without them. The inexperienced user wants to be led through the operation of the System; he does not, however, wish to be asked questions about optional employment of System functions with which he is not familiar. On the other hand, the experienced user wants to be able to marshal every last resource of the System. Finally, the inexperienced user should not be kept in a cocoon forever, and he must be at least given the opportunity to obtain an explanation of the various available features of the System.

##### III.1.1 The Query Sequence

The fundamental method of operation is embodied in the concept of a query sequence. Initially, the user sets up a retrieval command based on words. He is then given the opportunity to inspect the results of the retrieval, to modify the query or to discontinue the query sequence. During such a query sequence, a file of retrieved documents is built up. The three basic options available to the inexperienced user are:

- |       |  |
|-------|--|
| "END" | Terminate this search query sequence in order to start a new sequence or sign off.   |
| "MOD" | Modify or replace the present query and continue the present query sequence.   |
| "DOC" | Print data for documents retrieved during this sequence, or any documents of known accession number. The user is given a choice of the data to be printed. |

"HELP"      Not truly an option, this aids the user in the selection of the appropriate option name.

Ten other options exist, and those indicated by "\*" are actually entered automatically for the inexperienced user.

"OFF"	Generates file for printing bibliographic data and documents off-line.
"CHG"	Changes the mode of operation (sequence termination not required).
"CON"	Inspects the concept vectors of documents.
* "RET"	Executes the present retrieval request.
* "DEL"	Deletes unwanted documents retrieved during the present query sequence.
* "SEE"	Inspects the existing query.
* "CLR"	Erases the existing query.
* "WRD"	Adds or deletes query words.
* "DDC"	Performs document-document correlation.
* "WGT"	Performs direct manipulation of query concept vectors.

In addition to the options selected during a query sequence, a user may set various modes. The inexperienced user will take the default specification in which all modes are deselected, while the more experienced user may select one or more of the following:

1. Select terse dialogue.
2. Skip formation of initial query from words in query sequence.
3. Make available statistical analysis of query.
4. Make available statistical analysis of retrieval.
5. Assume sophisticated user.

Selection of the first mode results in terse, rather than verbose, messages being addressed from the System to the user. Most messages exist in two forms, and the terse form is used by experienced users. The second mode skips initial query formation, and lets the user select an option name immediately after signing onto the System.

If the third mode is selected, the user is asked if he wishes to see the words, stems and concept-weight pairs forming a word-based query. These data are available for printing before the query is executed. Also, he can elect the printing of the query concept vector itself. Although these options also exist under MOD, mode 3 makes them available for analysis of the initial query. Note that this mode does not cause the data to be printed, but simply gives the user the option of printing them.

Similarly, mode 4 is provided so that the user may be asked if he wants the contents of the temporary file (accession number, temporary identification number, rank and correlation when last retrieved, print suppression and whether or not the last executed retrieval retrieved the document) immediately after each retrieval. These data are otherwise available, but mode 4, like mode 3, provides a convenience for the serious student of the System.

Mode 5 simply causes "HELP" to result in the printing of the descriptions of all options, not just the basic three.

While building the temporary file, the user can delete irrelevant documents. Since the file is built up by the process of executing different retrieval requests, the re-retrieval of documents already retrieved once during the sequence may be inhibited at the user's choice.

If bibliographic data for a document have been printed once during a single query sequence, it is unlikely that the user will want these data printed again. Such printing is inhibited, but the user (even the inexperienced user) can override this inhibition.

If any words in the initial query are neither common nor found in the dictionary, and therefore do not enter into the retrieval process, they are listed for the user's information. If either none of the query words are in the dictionary or the query results in the retrieval of no documents, the user is so informed and asked to enter another query.

When a successful retrieval takes place, the user is told how many documents were retrieved. The accession numbers of the documents are placed in the temporary file. The System then, if the user so desires, starts to print more detailed information about the documents in the temporary file. For each document, the accession number and temporary identification number are printed. Then a check is made to see if bibliographic data for the document have been printed previously during the query sequence-- if not, the bibliographic data are printed. In the former case the output for a document occupies only a single line.

Clearly, users will infrequently want such data printed for the entire set of documents in the temporary file. On the other hand, in order to modify his query intelligently, the user must have some idea of what he has retrieved. After the data for five documents have been printed, the user is asked if more documents are wanted. If they are, five more are printed.

When either all the data for the documents in the temporary file have been printed or the user has decided he has seen enough, he is asked to enter an option name or, in order to get a brief explanation of the options, "HELP". A cry of "HELP" from the user results in the printing of descriptions of MOD, DOC and END options. Now, since it is not the desire to keep the inexperienced user from learning more about the System, he is asked if he wishes to see similar explanations of the remaining ten options, and if he does these are printed. (Similarly, if he attempts to use the option CHG, he is asked if he wishes to see a list of the modes available.)

The user is again asked to enter an option name. Although any legitimate option name will be accepted, the basic three will be used most frequently. An illegal option name will result in an error message and a request for an option name or "HELP", so that a user who misremembers a name is taken back to the point where aid is available.

The END option causes the user to be asked if he is through with the retrieval System. If he is, the System is shut down; if not, an entire new query sequence is initiated.

The DOC option allows the user to obtain more information about the documents presently in the temporary file, or any other documents for which the accession number is known. The user is first asked if he wants only bibliographic information for documents in the present temporary file, with information previously printed suppressed--just as results after an initial query. If he answers "YES", these data and the temporary file data are made available, with the question "MORE" following every five documents in the bibliographic section. It is expected that this would be done by a user who printed only a small part of the bibliographic data immediately following a retrieval and then wants to obtain more of it.

If the last-mentioned question is answered "NO", the user is asked to specify a document or document set of interest to him. He may do so by entering a single accession number or temporary identification number, or a range of temporary identification numbers, or the word "ALL" to signify all the documents in the temporary file. An illegal entry results in a more detailed explanation of the format required and a request that the user try again.

For each document specified, the accession number is first printed. If the document is in the temporary file, the following are printed: its temporary identification number, rank and correlation on its last retrieval, whether or not the last executed retrieval retrieved the document, and whether or not the bibliographic data for the document have already been printed.

If the document is suppressed from future retrieval, this fact is stated. Bibliographic data are printed if they have not been printed before; if they have, the operator is asked if they are to be printed again and appropriate action is taken. Next the operator is asked if the abstract is to be printed, and the System prints it in response to an answer of "YES".

If there was only one document specified by accession number or temporary identification, the user is given the opportunity to specify more. The process continues as above if he does, or requests an option name if he does not.

Printing an entire abstract may take some time, so even if a set of documents has been specified the user is asked if he wishes to continue after the printing of an abstract. Similarly, the user is asked if he wishes to continue after the printing of any information from five documents. A negative reply in either case results in a request for an option name, or the specification of other documents to be examined.

The MOD option not only allows the user to modify or replace his query, but it also automatically transfers the inexperienced user to sections of other options in order to delete\* entries from the temporary file (if desired or required) and perform retrieval\*\*. Upon entrance to MOD, the user is first asked if document-document correlation is to be used as the retrieval method. (Recall that he has started with query words and already retrieved some documents.)

If both document-document correlation is chosen and the last retrieval performed was also based on document-document correlation, the user is given the option of building on the concept vector used in the previous retrieval or starting afresh. He then builds or adds to a query vector by specifying any number of documents by means of single accession number,

---

\* DEL

\*\* RET

single or ranges of temporary identification numbers, or all the documents in the temporary file. After indicating that no more documents are to be used for the search, the user is asked if he desires to initiate the retrieval.

The point at which the user is asked about starting the retrieval can be reached by another path, which is started when the user rejects document-document correlation. The words forming the last query performed on a query word basis have been retained (with their stems and concept-weight mappings), so the user is given the choice of retaining and building on them or erasing them and building a new set of query words. The System is so designed that a user can inspect, modify and again inspect the set of query words, and so the user is asked if he wishes to inspect or modify the set or not. A negative answer causes the user to be asked if he wishes to initiate retrieval.

If the user indicates that he does wish to inspect or modify the set of query vector words, the present set (with stems and concept-weight pairs) is printed and he is then asked if he wants to add or replicate any words. If he does, he is asked to enter the words. Any noncommon, non-dictionary words are reported to the user if they are entered, and he is again given the chance to add or replicate words. The user is then given the opportunity to delete words, and informed if he attempts to delete any words not present and allowed to try again.

Next the user is given the opportunity to inspect the query concept vector directly, and if he so elects it is printed. He may add signed concept number-weight pairs, and is informed of any illegal concept numbers that he attempts to enter.

Use of the above three methods of query vector modification, or some combination of them, eventually leads the user to the point where he is asked if he wants a retrieval performed. It is possible that he wants to return to the point of entering an option name--for example, he might want

to have some additional document information printed, and then return to building a document-document correlation query. In such an event, he would answer the question about initiating retrieval in the negative.

When the user indicates that he does want to perform a retrieval, the dialogue processor determines if the query concept vector is null. If it is, the user has obviously become confused, and he is given the opportunity of either starting a new query sequence or resuming the present sequence with a new option name.

Assuming that a retrieval is requested and the query vector is not null, the user is informed if the temporary file is empty. He is asked to specify if documents previously retrieved during the query sequences are to be excluded from re-retrieval or not, and he is asked if printing of bibliographic data already printed once should be allowed or suppressed.

If the temporary file is full, the user is told that he must make space for the documents to be retrieved; if it is partially filled he is given the opportunity to delete documents. Documents to be deleted are specified by accession number, temporary identification number or range of temporary identification numbers. Alternatively, the entire temporary file may be deleted.

Then, in order that the user may identify contents of the temporary file with the particular queries retrieving them, he is informed of the starting temporary identification of the documents to be retrieved, and the retrieval is performed.

If no documents are retrieved, the user is so informed and asked to enter an option name or "HELP". If the retrieval is successful, the system continues just as it does after a successful initial retrieval.

### III.1.2     The Temporary File

Every time a retrieval is successfully executed during a query sequence, information concerning the documents retrieved is added to the temporary file, continuing until all retrieved documents have been placed in the file or until the file is full. The file capacity is 50 documents, but it may contain results of previous retrievals. Before a retrieval is executed, the user is informed that the file is presently empty, or informed of the remaining space and asked if additional space is required, or told that the file is full and that additional space must be created. If he retrieves more documents than there are spaces in the file, only the highest correlated documents are placed in the file.

During any query sequence, each retrieved document is assigned a temporary identification number. This number is used only for convenience, since it is potentially much shorter than the document's accession number. The user may need to specify a document for deletion from the temporary file, for the printing of bibliographic data or of the document itself, or for document-document correlation.

The temporary file contains only the following information:

1.     Accession number;
2.     Temporary identification number;
3.     Flag indicating if the last executed retrieval retrieved the document;
4.     Flag indicating if the bibliographic data for the document have been printed and the printing inhibition not removed;
5.     Correlation obtained during the last retrieval of the document;
6.     Rank obtained during the last retrieval of the document.

Subsection III. 2. 7 discusses the manner in which the temporary file is stored in detail.

In addition to the temporary file, there is a list of documents whose retrieval is excluded. These are documents which have been retrieved at least once during the retrieval process, that the user does not want to re-retrieve.

### III.1.3 Query Types

Initially, a set of query words is entered by the user.\* A file containing these words, their stems and weighted mapping into concepts is established. For additional retrievals during the query sequence, the file may be cleared and a new query entered. Or words may be deleted, added or replicated, building on the initial query.

After a retrieval, the query concept vector is retained. If the next retrieval is based on query words, the query concept vector is simply cleared and a new vector constructed from the query word file. The query word file itself may be entirely new or formed by adding and deleting words from the previous query word file. In the case of document-document correlation, the user may either build on the existing query concept vector or generate an entirely new one.

It is also possible for the user to manipulate the query concept vector directly.

### III.1.4 Levels of Document Information

Information concerning documents is available on three levels. First is the temporary file information, obviously available only for documents retrieved during the present query sequence. The only permanent information in the file is the document's accession number.

---

\* The experienced user may skip this activity.

There are also the bibliographic data, with such elements as author, title, date, etc. These data may be printed in a relatively short time, and the user may obtain them for either documents in the temporary file or for any other document whose accession number is known.

Finally, there are the documents themselves. These can be obtained in the same manner as the bibliographic data, and, of course, are comparatively lengthy. (In the presently contemplated data base, the "documents" are in fact abstracts of other documents.)

### III.1.5 Two Retrieval Dialogues

Figure III-1 contains two actual dialogues with the dialogue processor, as might result from use of the System by users of two different experience levels. For purposes of illustration, both users begin with the same query, and perform similar actions. The inexperienced user, whose query appears in Figure III-1(a), is guided extensively by the dialogue processor, and is not offered the display of various internal information that would only confuse him. On the other hand, the experienced user is permitted to display data that give great insight into the workings of the System. Thus, Figure III-1(a) emphasizes the tutorial operation of the system, while Figure III-1(b) shows in some detail the operation of the System.

III.1.5.1 Dialogue with Inexperienced User. This discussion refers to the dialogue of Figure III-1(a). This user knows what the System does, but has neither the need nor inclination to find out how the System works. He knows, for instance, that he should start a query by answering "yes" to the question, "Is normal operation desired?". This gives him the verbose form of all messages, and the simplest sequence of questions.

His initial query concerns information processing and information engineering. Three of the words in his query, "representation", "boradest", and "sense" are not in the stem dictionary, and are therefore not useful for retrieval from the collection. One of these, "boradest", is a

THIS IS THE RADC ON-LINE DIALOGUE PROCESSOR SIGNING ON.

IS NORMAL OPERATION DESIRED?:

= YES

ENTER WORDS FOR INITIAL SEARCH QUERY, FOLLOWED BY "END":

= THE INFORMATION PROCESSING REVOLUTION AND THE EMERGENCE OF

= THE PROFESSION OF INFORMATION ENGINEERING, DEALING WITH THE

= PROCESSING AND REPRESENTATION OF INFORMATION IN THE BORADEST SENSE END

THE FOLLOWING WORDS ARE NOT USEFUL FOR RETRIEVAL FROM THIS

COLLECTION:

REPRESENTATION

BORADEST

SENSE

THE NUMBER OF DOCUMENTS MEETING YOUR SPECIFICATIONS FOR THE

QUERY IS 31.

DO YOU WANT BIBLIOGRAPHIC INFORMATION FOR SOME OF THE RETRIEVED

-DOCUMENTS?:

= YES

ACC. NO. 49.

TEMP. ID. 31.

& DESCRIPTIVE & CATALOGING & GUIDE .

& U. & S. & ATOMIC & ENERGY & COMMISSION .

ACC. NO. 48.

TEMP. ID. 30.

& MANUAL FOR THE & ANALYSIS OF & LIBRARY & SYSTEMS .

& TAYLOR , & ROBERT & S. \* & HIEBER , & CAROLINE & E.

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

ACC. NO. 47.  
 TEMP. ID. 29.  
 & PUSH & BUTTON & BIBLIOGRAPHY .. & TODAY AND & TOMORROW .  
 & SHAFFER , & KENNETH & R. \* & SICKMAN , & LUDWIG \* & PARKER ,  
 & RALPH & H.

ACC. NO. 46.  
 -TEMP. ID. 28.  
 & BIBLIOGRAPHIC & INFORMATION & EXCHANGE .  
 & POPECKI , & JOSEPH & T.

ACC. NO. 43.  
 TEMP. ID. 27.  
 \*\*AUTHOR  
 & LAMKIN , & BURTON & E.  
 & LAMKIN , & BURTON & E.

MORE?:  
 = NO  
 ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE  
 OPTIONS):  
 = DOC  
 PRINT ONLY RANKING AND BIBLIOGRAPHIC DATA (NO ABSTRACTS) FOR  
 DOCUMENTS RETRIEVED DURING THIS QUERY SEQUENCE, EXCLUDING  
 BIBLIOGRAPHIC DATA ALREADY PRINTED?  
 = YES-

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

ACCESSION NUMBER	TEMPORARY IDENT.	CORRELATION [WHEN LAST RETRIEVED]	RANK	BIB. DATA PRINTED/ RET'D LAST QUERY
49	31	0.1414E+00	16	T/T
48	30	0.7036E-01	22	T/T
47	29	0.2917E+00	8	T/T
46	28	0.1741E+00	13	T/T
43	27	0.6244E-01	24	T/T
40	26	0.2958E-01	29	F/T
36	25	0.3659E-01	27	F/T
33	24	0.2428E+00	10	F/T
32	23	0.1387E+00	17	F/T
31	22	0.2572E+00	9	F/T
30	21	0.3257E+00	6	F/T
29	20	0.2946E-01	30	F/T
28	19	0.1462E+00	15	F/T
26	18	0.5390E-01	25	F/T
25	17 <sup>a</sup>	0.5285E-01	26	F/T
22	16	0.3769E+00	4	F/T
21	15	0.6510E-01	23	F/T
17	14	0.2357E-01	31	F/T
16	13	0.7087E-01	21	F/T
15	12	0.3658E+00	5	F/T
14	11	0.3113E+00	7	F/T
13	10	0.5595E+00	1	F/T
11	9	0.5280E+00	2	F/T
9	8	0.7180E-01	20	F/T
8	7	0.8165E-01	19	F/T
6	6	0.4857E+00	3	F/T
5	5	0.2157E+00	11	F/T
4	4	0.1591E+00	14	F/T
3	3	0.2000E+00	12	F/T
2	2	0.8980E-01	18	F/T
1	1	0.3178E-01	28	F/T

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

DO YOU WANT BIBLIOGRAPHIC INFORMATION FOR SOME OF THE RETRIEVED DOCUMENTS?:

- = NO

ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE OPTIONS):

- = DOC

PRINT ONLY RANKING AND BIBLIOGRAPHIC DATA (NO ABSTRACTS) FOR DOCUMENTS RETRIEVED DURING THIS QUERY SEQUENCE, EXCLUDING BIBLIOGRAPHIC DATA ALREADY PRINTED?

- = NO

SPECIFY FIRST DOCUMENT OR DOCUMENT GROUP TO PRINT.

ENTER TEMP ID. (SINGLE OR RANGE), ACC. NO. OR "ALL":

- = A13

ACC. NO. 13. ID. 10. CORR. 0.559. RANK 1. RETRIEVED BY LAST QUERY & WILL & THERE & BE A & PROFESSION OF & INFORMATION & ENGINEERING --- & CHUEY , & RICHARD & L.

--PRINT ABSTRACT?:

- = YES

IF THE ENGINEERING PROFESSION IS TO PLAY THE SAME ROLE IN THE CURRENT INFORMATION REVOLUTION THAT IT PLAYED IN THE INDUSTRIAL REVOLUTION , IT MUST BOTH MODIFY ITSELF AND MERGE WITH OTHER PROFESSIONS , IN ORDER TO ASSIMILATE KNOWLEDGE OF , AND ACQUIRE THE ABILITY TO DESIGN , SYSTEMS IN WHICH THE PRIMARY COMMODITY IS INFORMATION & NOT ENERGY OR MATERIAL ] . WHETHER THE MEN RESPONSIBLE FOR CREATING FUTURE INFORMATION SYSTEMS WILL CONSIDER THEMSELVES MEMBERS OF THE ENGINEERING PROFESSION REMAINS TO BE SEEN . Z \$ CLB J

MORE?:

- = NO

ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE OPTIONS):

- = HELP

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

OPTIONS AVAILABLE ARE:

"END" - TERMINATE THIS SEARCH QUERY SEQUENCE FOR STARTING  
A NEW SEQUENCE OR SIGNING OFF.

"MOD" - MODIFY OR REPLACE THE PRESENT QUERY AND CONTINUE  
THE PRESENT QUERY SEQUENCE.

"DOC" - PRINT DATA FOR DOCUMENTS RETRIEVED DURING THIS  
SEQUENCE OR ANY DOCUMENTS OF KNOWN ACCESSION NUMBER.

OTHER OPTIONS ARE AVAILABLE. DO YOU WANT A LIST?

- NO  
ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE  
OPTIONS):

- MOD  
DO YOU WANT TO ERASE THE PRESENT QUERY AND DO DOCUMENT-DOCUMENT  
SEARCHING?

- YES

NOW SPECIFY THE DOCUMENTS FOR CORRELATION.

ENTER TEMP ID. (SINGLE OR RANGE), ACC. NO. OR "ALL":

- ALL

MORE?

- NO

DO YOU WANT TO INSPECT OR DIRECTLY MODIFY THE QUERY CONCEPT  
VECTOR?

- NO

DO YOU WANT A RETRIEVAL PERFORMED WITH THE PRESENT QUERY VECTOR?

- YES

SHOULD DOCUMENTS RETRIEVED PREVIOUSLY DURING THIS QUERY SEQUENCE  
BE EXCLUDED FROM RE-RETRIEVAL?

- NO

SHOULD PRINTING OF BIBLIOGRAPHIC DATA PREVIOUSLY PRINTED BE  
SUPPRESSED?

- YES

THERE ARE 19 SPACES IN THE TEMPORARY FILE.  
 SPACES EXIST IN THE TEMPORARY FILE FOR NEW RETRIEVALS. IS MORE  
 SPACE DESIRED?:  
 = YES  
 SOME DOCUMENTS ARE TO BE DELETED FROM THE TEMPORARY FILE--  
 BEFORE THE PRESENT RETRIEVAL IS PERFORMED.  
 YOU MUST SELECT THE DOCUMENTS TO BE DELETED.  
 ENTER TEMP ID. (SINGLE OR RANGE), ACC. NO. OR "ALL":  
 = ALL  
 DOCUMENTS FOUND BY THIS RETRIEVAL WILL HAVE TEMP. NOS.  
 STARTING WITH 32.  
 THE NUMBER OF DOCUMENTS MEETING YOUR SPECIFICATIONS FOR THE  
 QUERY IS 44.  
 DO YOU WANT BIBLIOGRAPHIC INFORMATION FOR SOME OF THE RETRIEVED  
 DOCUMENTS?:  
 = NO  
 ENTER OPTION NAME (OR LP" TO SEE THE LIST OF AVAILABLE  
 OPTIONS):  
 = DOC  
 PRINT ONLY RANKING AND BIBLIOGRAPHIC DATA (NO ABSTRACTS) FOR  
 DOCUMENTS RETRIEVED DURING THIS QUERY SEQUENCE, EXCLUDING  
 BIBLIOGRAPHIC DATA ALREADY PRINTED?  
 = YES

ACCESSION NUMBER	TEMPORARY IDEN.	CORRELATION (WHEN LAST RETRIEVED)	RANK [WHEN LAST RETRIEVED]	BIB. DATA PRINTED/ RET'VD LAST QUERY
49	75	0.1187E+00	18	F/T
48	74	0.1535E+00	12	F/T
47	73	0.2571E+00	6	F/T
46	72	0.1461E+00	16	F/T
45	71	0.1088E-01	44	F/T
44	70	0.2575E-01	36	F/T
43	69	0.3144E-01	34	F/T
41	68	0.5843E-01	29	F/T

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

40	0.1489E-01	41	F/T
39	0.1465E+00	15	F/T
38	0.2806E-01	35	F/T
37	0.9174E-01	20	F/T
36	0.1842E-01	40	F/T
35	0.4263E-01	32	F/T
34	0.6924E-01	25	F/T
33	0.1334E+00	17	F/T
32	0.9310E-01	19	F/T
31	0.1727E+00	9	F/T
30	0.2343E+00	7	F/T
29	0.7417E-01	24	F/T
28	0.8832E-01	21	F/T
27	0.3707E-01	33	F/T
26	0.6106E-01	27	F/T
25	0.6209E-01	26	F/T
24	0.4350E-01	31	F/T
23	0.1164E-01	43	F/T
22	0.3289E+00	4	F/T
21	0.5463E-01	30	F/T
19	0.1327E-01	42	F/T
17	0.2374E-01	37	F/T
16	0.3028E-01	23	F/T
15	0.3575E+00	3	F/T
14	0.2090E+00	8	F/T
13	0.1000E+01	1	F/T
11	0.3706E+00	2	F/T
10	0.2006E-01	39	F/T
9	0.1687E+00	10	F/T
8	0.8222E-01	22	F/T
7	0.2050E-01	38	F/T
6	0.2890E+00	5	F/T
5	0.1629E+00	11	F/T
4	0.1469E+00	14	F/T
3	0.1511E+00	13	F/T
2	0.6029E-01	28	F/T

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

DO YOU WANT BIBLIOGRAPHIC INFORMATION FOR SOME OF THE RETRIEVED  
DOCUMENTS?  
= YES

ACC. NO. 49.  
TEMP. ID. 75.  
& DESCRIPTIVE & CATALOGING & GUIDE .  
& U. & S. & ATOMIC & ENERGY & COMMISSION .

ACC. NO. 48.  
TEMP. ID. 74.  
& MANUAL FOR THE & ANALYSIS OF & LIBRARY & SYSTEMS .  
& TAYLOR , & ROBERT & S. \* & HIEBER , & CAROLINE & E.

ACC. NO. 47.  
TEMP. ID. 73.  
& PUSH & BUTTON & BIBLIOGRAPHY ... & -TODAY AND & TOMORROW .  
& SHAFFER , & KENNETH & R. \* & SICKMAN , & LUDWIG \* & PARKER ,  
& RALPH & H.

ACC. NO. 46.  
TEMP. ID. 72.  
& BIBLIOGRAPHIC & INFORMATION & EXCHANGE .  
& POPECKI , & JOSEPH & T.

ACC. NO. 45.  
TEMP. ID. 71.  
& BOOK & CATALOGS VERSUS & CARD & CATALOGS .  
& PIZER , & IRWIN & H.

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

MORE?:  
 = NO  
 ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE  
 OPTIONS):  
 = D0C  
 PRINT ONLY RANKING AND BIBLIOGRAPHIC DATA (NO ABSTRACTS) FOR  
 DOCUMENTS RETRIEVED DURING THIS QUERY SEQUENCE, EXCLUDING  
 BIBLIOGRAPHIC DATA ALREADY PRINTED?  
 = NO  
 SPECIFY FIRST DOCUMENT OR DOCUMENT GROUP TO PRINT.  
 ENTER TEMP ID. (SINGLE OR RANGE), ACC. NO. OR "ALL":  
 = A11

ACC. NO. 11. ID. 41. CORR. 0.371. RANK 2. RETRIEVED BY LAST QUERY  
 & THE & ART OF & TEACHING & INFORMATION & SCIENCE .  
 & REES , & ALAN & M.  
 PRINT ABSTRACT?:  
 = YES

IN THE ABSENCE OF AN ADEQUATE AND AGREED DEFINITION OF INFORMATION  
 SCIENCE , TEACHING MUST ESSENTIALLY BE BASED UPON A HIGHLY SUBJECTIVE  
 VIEWPOINT . THE HETEROGENEOUS ORIGINS OF WORKERS IN THE FIELD OF  
 INFORMATION SCIENCE--MATHEMATICS , LINGUISTICS , LIBRARY SCIENCE ,  
 LOGIC , PSYCHOLOGY , COMPUTER TECHNOLOGY , ETC.--FURTHER CONTRIBUTE  
 TO A LACK OF COMMON AGREEMENT AS TO THE PARAMETERS OF THE FIELD . THE  
 LACK OF ADEQUATE TEXTBOOKS IS BOTH A RESULT AND A CAUSE OF THE  
 SHIFTING DEFINITION OF THE FIELD .

THE SUBJECTIVE APPROACH IS PREDOMINANT IN TEACHING . THE TEACHERS OF  
 INFORMATION SCIENCE ARE IN MANY INSTANCES THE 00 STARS 00 OF THE  
 FIELD WHO HAVE ACHIEVED EMINENCE AND ACCLAIM FOR THEIR INVENTIVENESS  
 AND INGENUITY IN SYSTEMS DESIGN , OPERATION , MANAGEMENT AND  
 EXPERIMENTATION . COURSES CONSTITUTE THE SUM TOTAL OF AN INDIVIDUAL'S  
 ESSENTIALLY PAROCHIAL VIEWPOINT AND ARE ANALOGOUS TO THE HIGHLY  
 INDIVIDUALISTIC INTERPRETATION OF ARTISTICS PERFORMERS . 00 ONE-SEMESTER  
 STANDS , 00 WITH LITTLE INTEGRATION WITH THE REST OF THE CURRICULUM  
 OF THE TEACHING INSTITUTION , ARE QUITE COMMON .

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

THE AUTHOR'S EXPERIENCE IN APPROACHING THE TEACHING OF INFORMATION SCIENCE WITH AN ACADEMIC AND ELECTRIC APPROACH AT A WESTERN & RESERVE UNIVERSITY AND ELSEWHERE IS DESCRIBED IN DETAIL. THE DESIGN AND TEACHING OF TWO COURSES, 00 & INTRODUCTION TO & INFORMATION & RETRIEVAL & SYSTEMS 00 AND 00 & INFORMATION & CENTERS AND & INFORMATION & SERVICES, 00 ARE ANALYZED. TOPICS DISCUSSED INCLUDE .. DELINEATION OF THE 00 INFORMATION PROBLEM 00 .. STRUCTURE AND ANALYSIS OF INFORMATION SYSTEMS .. PARACHIAL AND EXTERNAL RESEARCH .. RESEARCH METHODOLOGY IN INFORMATION SCIENCE .. SYNTHESIS OF RESEARCH AND OPERATIONAL ACTIVITIES. THE MATTER OF TIMING AND SEQUENCE OF PRESENTATION IN THE TEACHING OF AN INTERDISCIPLINARY SUBJECT TO STUDENTS WITH INTERDISCIPLINARY BACKGROUNDS IS ILLUSTRATED. THE USE OF GUEST LECTURES IS DESCRIBED IN TERMS OF SHOWING AN APPLICATION OF GENERAL PRINCIPLES TO SPECIFIC SITUATIONS. X \$ AUTHOR ] MORE?:

= YES

ENTER TEMP ID. (SINGLE OR RANGE), ACC. NO. OR "ALL":

= A15

ACC. NO. 15. ID. 44. CORR. 0.358. RANK 3. RETRIEVED BY LAST QUERY & TOWARD AN & EDUCATIONAL & BASE FOR THE & INFORMATION & SCIENCES AND & INFORMATION & ENGINEERING.

& TAYLOR, & ROBERT & S.

PRINT ABSTRACT?:

= YES

THIS PAPER DEFINES AND DISCUSSES THE EDUCATION AND COMPETENCES FOR TWO MAJOR AREAS OF AN AS YET UNNAMED SUBJECT, WHICH IN THIS PAPER IS LABELED 00 & SUBJECT & X. 00 THIS 00 & SUBJECT & X 00 PERTAINS TO THE THEORETICAL, EXPERIMENTAL, AND OPERATIONAL STUDY OF THE INTERFACE BETWEEN MAN AND SYSTEMATIZED KNOWLEDGE. THE TWO AREAS ARE INFORMATION ENGINEERING AND THE INFORMATION SCIENCES. THE FORMER IS CONCERNED WITH THE DEVELOPMENT OF OPERATING SYSTEMS. THE LATTER, WITH THE EXPLICATION OF SYSTEMS AND THEIR COMPONENTS. FOR A VIABLE AND EFFECTIVE ACADEMIC DISCIPLINE, IT IS NECESSARY TO ESTABLISH AN ENVIRONMENT FOR FRUITFUL DIALOGUE BETWEEN OPERATIONAL PERSONNEL AND

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

RESEARCH PERSONNEL . THIS DOES NOT OCCUR , PRIMARILY BECAUSE THE STUDY OF OPERATING INFORMATION SYSTEMS HAS NOT DEVELOPED A FORMAL SET OF TOOLS AND SYMBOLS BY WHICH THESE PROCESSES CAN BE QUANTITATIVELY DESCRIBED . X & AUTHOR ]  
 MORE?:  
 = NO  
 ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE OPTIONS):  
 = MOD  
 DO YOU WANT TO PERFORM MORE DOCUMENT-DOCUMENT SEARCHING?:  
 = NO  
 DO YOU WANT TO ERASE COMPLETELY YOUR PRESENT QUERY AND ENTER NEW QUERY WORDS?:  
 = NO  
 DO YOU WANT TO SEE OR MODIFY THE WORDS FORMING THE QUERY?:  
 = SEE  
 ANSWER "YES" OR "NO":  
 = NO  
 DO YOU WANT TO ADD OR REPLICATE ANY WORDS?:  
 = ADD  
 ANSWER "YES" OR "NO":  
 = YES  
 ENTER WORDS, FOLLOWED BY "END":  
 : TEACHING OF INFORMATION SCIENCE. THE INTERFACE BETWEEN MAN AND  
 : SYSTEMATIZED KNOWLEDGE. THE SYNTHESIS AND ANALYSIS OF INFORMATION  
 : SYSTEMS. THE LOGICAL FOUNDATIONS OF INFORMATION SCIENCE AND  
 : INFORMATION ENGINEERING.  
 THE FOLLOWING WORDS ARE NOT USEFUL FOR RETRIEVAL FROM THIS  
 COLLECTION:  
 FOUNDATIONS  
 DO YOU WANT TO DELETE ANY WORDS?:  
 = NO  
 DO YOU WANT TO ERASE COMPLETELY YOUR PRESENT QUERY AND ENTER NEW QUERY WORDS?:  
 = NO

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

DO YOU WANT TO SEE OR MODIFY THE WORDS FORMING THE QUERY?:  
 = NO-0  
 DO YOU WANT TO INSPECT OR DIRECTLY MODIFY THE QUERY CONCEPT VECTOR?:  
 = NO  
 DO YOU WANT A RETRIEVAL PERFORMED WITH THE PRESENT QUERY VECTOR?:  
 = YES  
 SHOULD DOCUMENTS RETRIEVED PREVIOUSLY DURING THIS QUERY SEQUENCE BE EXCLUDED FROM RE-RETRIEVAL?:  
 = NO  
 SHOULD PRINTING OF BIBLIOGRAPHIC DATA PREVIOUSLY PRINTED BE SUPPRESSED?:  
 = YES  
 THERE ARE 6 SPACES IN THE TEMPORARY FILE.  
 SPACES EXIST IN THE TEMPORARY FILE FOR NEW RETRIEVALS. IS MORE SPACE DESIRED?:  
 = YES  
 SOME DOCUMENTS ARE TO BE DELETED FROM THE TEMPORARY FILE-- BEFORE THE PRESENT RETRIEVAL IS PERFORMED.  
 YOU MUST SELECT THE DOCUMENTS TO BE DELETED.  
 ENTER TEMP ID. (SINGLE OR RANGE), ACC. NO. OR "ALL":  
 = ALL  
 DOCUMENTS FOUND BY THIS RETRIEVAL WILL HAVE TEMP. NOS. STARTING WITH 76.  
 THE NUMBER OF DOCUMENTS MEETING YOUR SPECIFICATIONS FOR THE QUERY IS 35.  
 DO YOU WANT BIBLIOGRAPHIC INFORMATION FOR SOME OF THE RETRIEVED DOCUMENTS?:  
 = NO  
 ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE OPTIONS):  
 = D0C  
 PRINT ONLY RANKING AND BIBLIOGRAPHIC DATA (NO ABSTRACTS) FOR DOCUMENTS RETRIEVED DURING THIS QUERY SEQUENCE, EXCLUDING BIBLIOGRAPHIC DATA ALREADY PRINTED?  
 = YES

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

ACCESSION NUMBER	TEMPORARY IDENT.	CORRELATION [WHEN LAST RETRIEVED]	RANK	BIB. DATA PRINTED/ RET'D LAST QUERY
49	110	0.1461E+00	15	F/T
48	109	0.1272E+00	17	F/T
47 <sup>a</sup>	108	0.3013E+00	9	F/T
46	107	0.1798E+00	14	F/T
44	106	0.1980E-01	32	F/T
43	105	0.2418E-01	31	F/T
41	104	0.4495E-01	28	F/T
39	103	0.1127E+00	18	F/T
38	102	0.1439E-01	35	F/T
37	101	0.6049E-01	25	F/T
35	100	0.3279E-01	30	F/T
34	99	0.5326E-01	27	F/T
33	98	0.2223E+00	11	F/T
31	97	0.3321E+00	8	F/T
30	96	0.3484E+00	7	F/T
28	95	0.9058E-01	19	F/T
26	94	0.4175E-01	29	F/T
25	93	0.5459E-01	26	F/T
22	92	0.3892E+00	6	F/T
21	91	0.6723E-01	24	F/T
16	90	0.6862E-01	23	F/T
15	89	0.5500E+00	2	F/T
14	88	0.4501E+00	4	F/T
13	87	0.4767E+00	3	F/T
12	86	0.6870E-01	22	F/T
11	85	0.5577E+00	1	F/T
10	84	0.1543E-01	34	F/T
9	83	0.7415E-01	21	F/T
8	82	0.8433E-01	20	F/T
7	81	0.1577E-01	33	F/T
6	80	0.4104E+00	5	F/T
5	79	0.2227E+00	10	F/T
4	78	0.1849E+00	13	F/T
3	77	0.2066E+00	12	F/T
2	76	0.1391E+00	16	F/T

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

DO YOU WANT BIBLIOGRAPHIC INFORMATION FOR SOME OF THE RETRIEVED DOCUMENTS?  
 = NO  
 ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE OPTIONS):  
 = D0C  
 PRINT ONLY RANKING AND BIBLIOGRAPHIC DATA (NO ABSTRACTS) FOR DOCUMENTS RETRIEVED DURING THIS QUERY SEQUENCE, EXCLUDING BIBLIOGRAPHIC DATA ALREADY PRINTED?  
 = NO  
 SPECIFY FIRST DOCUMENT OR DOCUMENT GROUP TO PRINT.  
 ENTER TEMP ID. (SINGLE OR RANGE), ACC. NO. OR "ALL":  
 = A14  
 ACC. NO. 14. ID. 83. CORR. 0.450. RANK 4. RETRIEVED BY LAST QUERY & ON THE & NATURE OF & INFORMATION & SCIENCE AND THE & RESPONSIBILITY OF & INSTITUTIONS OF & HIGHER & EDUCATION .  
 & SLAMECKA , & VLADIMIR .  
 PRINT ABSTRACT?:  
 = YES  
 INFORMATION SCIENCE IS DESCRIBED AS AN INTERDISCIPLINARY FIELD CONCERNED WITH THE NATURE , PROPERTIES , CONTROL , AND USE OF INFORMATION . THIS INTERDISCIPLINARY NATURE HAS AN IMPORTANT BEARING ON EDUCATION IN THE FIELD ., THE TIME REQUIRED TO ATTAIN A SCHOLARLY OR RESEARCH LEVEL IS LONGER THAN IN MORE HOMOGENEOUS AREAS , SUCH AS MATHEMATICS . INSTITUTIONS OF HIGHER EDUCATION ARE , THEREFORE , IN GOOD POSITION TO SUPPORT RESEARCH AND EDUCATION IN INFORMATION SCIENCE BECAUSE THEY POSSESS CAPABILITIES IN THE MANY DISCIPLINES WHICH COMPRISE THE FIELD . PROGRAMS IN INFORMATION SCIENCE SHOULD BE ORGANIZED SO AS TO UTILIZE FULLY THE RESOURCES AND CAPABILITIES OF THE ENTIRE INSTITUTION . Z & AUTHOR 1  
 MORE?:  
 = NO

Figure III-1(a) Dialogue with Inexperienced User (cont'd)

ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE  
OPTIONS):

= END

PRESENT SEARCH QUERY SEQUENCE TERMINATED. A NEW QUERY MAY  
BE INITIATED AT THIS TIME OR YOU MAY SIGN OFF.

DO YOU WISH TO CONTINUE IN THE SAME MODE?:

= NO

DO YOU WISH TO TERMINATE USE OF THE SYSTEM?:

= YES

ON-LINE RETRIEVAL SYSTEM SIGNING OFF.

Figure III-1(a) Dialogue with Inexperienced User (concluded)

misspelling of "broadest", but for simplicity, he is not asked if he wants to change his query; an experienced user would be able to correct his spelling at this point, by modifying his query.

The processor performs a retrieval, placing 31 documents in the temporary file. The user indicates that he would like bibliographic information for some documents; he is given this for five documents. He then decides, on the basis of this data, that he would like to see the entire ranking table of documents retrieved. He enters option "DOC" to print this query. To print selected documents from the table, he again enters option "DOC". He bypasses printing of the table to print selectively. The title, "Will there be a profession of information engineering?" appears to be very similar to his query, so he prints the abstract. Satisfied that this document is very nearly what he wanted, he decides to find others like it. But he has used only option "DOC", and does not know how to perform document-document searching. His cry of "HELP" produces a list of the options he can select at this point. Seeing that he can use "MOD", he suppresses printing of further options.

The highest-ranked document alone is used for document-document searching, without modifying the query vector. The user does not exclude from re-retrieval documents retrieved during this sequence; he wishes to observe changes in the ranking, and use this information to guide his browsing. But he will not require re-printing of bibliographic data that he has in front of him, so he suppresses it. Since he has only one document he knows to be of interest, he deletes all entries from the temporary file. Following the retrieval, he prints the ranking table, entering option "DOC" to do so.

From the table, note that document A13 (accession number 13) correlates with itself with a correlation of 1.0, as would be expected. Document A11 has remained second-ranked, while A15 has risen from fifth to third. The user decides to print bibliographic data for these. After obtaining five sets of bibliographic data, he enters option "DOC" to print selected data. He finds both A11 and A15 of interest, and decides to modify his query, adding material from these documents.

After entering option "MOD" to change his query vector, the user incorrectly answers "SEE" to indicate that he wants to see his query words. The System corrects him, and asks him to answer again. Because no words were entered for this query (because document-document searching was performed) there are presently no words in the query. An experienced user would also be able to list the concept numbers and weights in his query. He adds to his query words dealing with information science. He then performs another retrieval. Once again, because of the small size of the collection, he clears the temporary table before retrieving.

He once again enters option "DOC" to print the ranking table. In this table, A13 is no longer top-ranked; the changes have made the query more like A11 than A13. Documents A15 and A13, which are now second and third, have already been printed; so the user decides to inspect A14, which in three retrievals has ranked seventh, eighth, and fourth. The bibliographic data confirms his interest, and he prints the abstract.

The user now has found four apparently relevant documents. At this point, he would probably look at the actual documents, to make a final relevance judgment. Then, if he was not completely satisfied with these four, he might initiate another query sequence. Thus, in addition to the browsing that takes place during a query sequence as illustrated by this example, there could exist another higher level of browsing, as the user converged upon the desired documents by successive query sequences, alternating with inspecting documents.

III.1.5.2 Dialogue with More Experienced User. Figure III-1(b) shows a dialogue that might be conducted by a more experienced user, who has used the System several times and who was becoming proficient in its use. For purposes of this discussion, the initial query entered by this user is the same as the one entered in the dialogue of Figure III-1(a).

IS NORMAL OPERATION DESIRED?:  
 = NO  
 DO YOU WANT AN EXPLANATION OF THE AVAILABLE MODES?:  
 = NO  
 MODE FLAGS ALL OFF. IDENTIFY NUMBERS OF FLAGS TO BE SET ON,  
 FOLLOWED BY "END":  
 = WHAT?  
 ILLEGAL SELECTION; REQUEST IGNORED.  
 DO YOU WANT AN EXPLANATION OF THE AVAILABLE MODES?:  
 = YES  
 MODES ARE NORMALLY "OFF" AND CAN BE TURNED ON BY TYPING IN A FLAG  
 NUMBER OR SEQUENCE OF NUMBERS, SUCH AS "1,3,5,END". THE FOLLOWING  
 MODES ARE AVAILABLE:

FLAG NUMBER	ACTION
1	SELECT TERSE DIALOGUE.
2	SKIP FORMATION OF INITIAL QUERY IN QUERY SEQUENCE FROM WORDS.
3	MAKE AVAILABLE QUERY WORDS, STEMS AND CONCEPTS BEFORE RETRIEVAL.
4	MAKE AVAILABLE TEMP TABLE CONTENTS AFTER RETRIEVAL.
5	ASSUME ANY OPTION MAY BE USED.

MODE FLAGS ALL OFF. IDENTIFY NUMBERS OF FLAGS TO BE SET ON,  
 FOLLOWED BY "END":  
 = 1,3,4,END  
 OPTIONS:  
 = HELP  
 OPTIONS: "END", "MOD", "DOC".  
 MORE?:  
 = YES  
 "OFF", "CHG", "CON", "RET", "DEL", "SEE", "CLR", "WRD", "DDC", "WGT".  
 OPTIONS:  
 = CHG

Figure III-1(b) Dialogue With Experienced User (cont'd)

DO YOU WANT AN EXPLANATION OF THE AVAILABLE MODES?:

- = NO
- MODE FLAGS ALL OFF. IDENTIFY NUMBERS OF FLAGS TO BE SET ON, FOLLOWED BY "END":
- = END
- ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE OPTIONS):
- = HELP
- OPTIONS AVAILABLE ARE:

"END" - TERMINATE THIS SEARCH QUERY SEQUENCE FOR STARTING A NEW SEQUENCE OR SIGNING OFF.

"MOD" - MODIFY OR REPLACE THE PRESENT QUERY AND CONTINUE THE PRESENT QUERY SEQUENCE.

"DOC" - PRINT DATA FOR DOCUMENTS RETRIEVED DURING THIS SEQUENCE OR ANY DOCUMENTS OF KNOWN ACCESSION NUMBER.

OTHER OPTIONS ARE AVAILABLE. DO YOU WANT A LIST?:

- = YES
- "OFF" - CREATE FILE FOR OFFLINE DOCUMENT PRINTING
- "CHG" - CHANGE MODE OF OPERATION (QUERY SEQUENCE TERMINATION NOT REQUIRED.) A LIST OF MODES IS PROVIDED.
- "CON" - INSPECT THE CONCEPT VECTORS OF DOCUMENTS.
- "RET" - EXECUTE THE PRESENT RETRIEVAL REQUEST.
- "DEL" - DELETE UNWANTED DOCUMENTS RETRIEVED DURING THE PRESENT QUERY SEQUENCE.
- "SEE" - INSPECT THE EXISTING QUERY.
- "CLR" - ERASE THE EXISTING QUERY.
- "WRD" - ADD OR DELETE QUERY WORDS.
- "DDC" - PERFORM DOCUMENT-DOCUMENT CORRELATION.
- "WGT" - PERFORM DIRECT MANIPULATION OF QUERY CONCEPT VECTORS.

(OPTIONS MARKED WITH "\*" ARE NORMALLY CALLED AUTOMATICALLY FOR THE USER BY THE SYSTEM.)

Figure III-1(b) Dialogue With Experienced User (cont'd)

```

ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE
OPTIONS):
= CHG
DO YOU WANT AN EXPLANATION OF THE AVAILABLE MODES?:
= NO
MODE FLAGS ALL OFF. IDENTIFY NUMBERS OF FLAGS TO BE SET ON,
FOLLOWED BY "END":
= 1,3,4,END
OPTIONS:
= MOD
DOC.-DOC.?:
= NO
TOTALLY REPLACE PRESENT QUERY?:
= NOOOOYES
ENTER WORDS:
= THE INFORMATION PROCESSING REVOLUTION AND THE EMERGENCE OF THE
= PROFESSION OF INFORMATION ENGINEERING, DEALING WITH THE PROCESSING
= AND REPRESENTATION OF INFORMATION IN THE BORADEST SENSE. END
WORDS NOT IN DICTIONARY:
REPRESENTATION
BORADEST
SENSE
DELETE?:
= NO
TOTALLY REPLACE PRESENT QUERY?:
= NO
QUERY WORD ACTION?:
= YES
PRESENT:
INFORMATION
PROCESSING
REVOLUTION
EMERGENCE
PROFESSION
INFORMATION
ENGINEERING
DEALING
PROCESSING
INFORMATION

```

Figure III-1(b) Dialogue With Experienced User (cont'd)

```

ADD WORDS?:
= YES
ENTER WORDS:
= BROADEST
= END
WORDS NOT IN DICTIONARY:
BROADEST
DELETE?:
= NO
TOTALLY REPLACE PRESENT QUERY?:
= NO
QUERY WORD ACTION?:
= NO
DIRECT CON. VECT. ACTION?:
= YES

CON- WEIGHT CON- WEIGHT CON- WEIGHT CON- WEIGHT
CEPT CEPT CEPT CEPT CEPT
0 0. 0 0. 0 0. 209 1.00000 283 1.00000
653 1.00000 272 1.00000 727 1.00000 648 2.00000 421 3.00000

MODIFY?:
= NO
RETRIEVE?:
= YES
TEMP. FILE EMPTY TO START.
EXCLUDE PREVIOUS?:
= NO
SUPPRESS PREV. PRINTED?:
= NO
TEMP ID. STARTS WITH 1.
PRINT PRSNT?:
= YES

```

Figure III-1(b) Dialogue With Experienced User (cont'd)

WORD	STEM	CONCEPT-WEIGHT PAIRS	
INFORMATION	INFORMA	421 1.00000	0 0.
PROCESSING	PROCE	648 1.00000	0 0.
REVOLUTION	REVOLU	727 1.00000	0 0.
EMERGENCE	EMERG	272 1.00000	0 0.
PROFESSION	PROFE	653 1.00000	0 0.
INFORMATION	INFORMA	421 1.00000	0 0.
ENGINEERING	ENGINE	283 1.00000	0 0.
DEALING	DEALING	209 1.00000	0 0.
PROCESSING	PROCE	648 1.00000	0 0.
INFORMATION	INFORMA	421 1.00000	0 0.

PRINT QUERY VECTOR??

■ NO  
 NO. OF HITS= 31.  
 PRINT TEMP??  
 ■ YES

Figure III-1(b) Dialogue With Experienced User (cont'd)

ACCESSION NUMBER	TEMPORARY IDENT.	CORRELATION [WHEN LAST RETRIEVED]	RANK	BIB. DATA PRINTED/ RET'D LAST QUERY
49	31	0.1414E+00	16	F/T
48	30	0.7036E-01	22	F/T
47	29	0.2917E+00	8	F/T
46	28	0.1741E+00	13	F/T
43	27	0.6244E-01	24	F/T
40	26	0.2958E-01	29	F/T
36	25	0.3659E-01	27	F/T
33	24	0.2428E+00	10	F/T
32	23	0.1387E+00	17	F/T
31	22	0.2572E+00	9	F/T
30	21	0.3257E+00	6	F/T
29	20	0.2946E-01	30	F/T
28	19	0.1462E+00	15	F/T
26	18	0.5390E-01	25	F/T
25	17	0.5285E-01	26	F/T
22	16	0.3769E+00	4	F/T
21	15	0.6510E-01	23	F/T
17	14	0.2357E-01	31	F/T
16	13	0.7087E-01	21	F/T
15	12	0.3658E+00	5	F/T
14	11	0.3113E+00	7	F/T
13	10	0.5595E+00	1	F/T
11	9	0.5280E+00	2	F/T
9	8	0.7180E-01	20	F/T
8	7	0.8165E-01	19	F/T
6	6	0.4857E+00	3	F/T
5	5	0.2157E+00	11	F/T
4	4	0.1591E+00	14	F/T
3	3	0.2000E+00	12	F/T
2	2	0.8980E-01	18	F/T
1	1	0.3178E-01	28	F/T

Figure III-1(b) Dialogue With Experienced User (cont'd)

```

PRINT BIBLIO.?:
= NO
OPTIONS:
= DDC
TEMP. DOCS. ONLY, SHORT FORM?:
= NO

IDENTIFY DOCUMENTS:
= A13

ACC. NO. 13. ID. 10. CORR. 0.559. RANK 1. RETRIEVED BY LAST QUERY
& WILL & THERE BE A & PROFESSION OF & INFORMATION & ENGINEERING .---
& SHUEY , & RICK/RD & L.
PRINT ABSTRACT?:
= YES
IF THE ENGINEERING PROFESSION IS TO PLAY THE SAME ROLE IN THE
CURRENT INFORMATION REVOLUTION THAT IT PLAYED IN THE INDUSTRIAL
REVOLUTION , IT MUST BOTH MODIFY ITSELF AND MERGE WITH OTHER
PROFESSIONS , IN ORDER TO ASSIMILATE KNOWLEDGE OF , AND ACQUIRE
THE ABILITY TO DESIGN , SYSTEMS IN WHICH THE PRIMARY COMMODITY IS
INFORMATION & NOT ENERGY OR MATERIAL . WHETHER THE MEN RESPONSIBLE
FOR CREATING FUTURE INFORMATION SYSTEMS WILL CONSIDER THEMSELVES
MEMBERS OF THE ENGINEERING PROFESSION REMAINS TO BE SEEN . X $ CLB J
MORE?:
= NO
OPTIONS:
= DDC
DOC.-DOC.?:
= YES

IDENTIFY DOCUMENTS:
= A13
MORE?:
= NO

```

Figure III-1(b) Dialogue With Experienced User (cont'd)

**STY -**

C&N- CEPT	WEIGHT	C&N- PT	WEIGHT	C&N- CEPT	WEIGHT	C&N- CEPT	WEIGHT	C&N- CEPT	WEIGHT
0	0.	813	0.50000	748	0.25000	733	0.25000	729	0.25000
727	0.50000	721	0.25000	705	0.25000	654	0.25000	653	0.75000
641	0.25000	620	0.25000	619	0.25000	583	0.25000	547	0.25000
536	0.25000	534	0.25000	533	0.25000	518	0.25000	463	0.25000
461	0.25000	421	1.00000	419	0.25000	358	0.25000	283	0.75000
282	0.25000	227	0.25000	203	0.25000	201	0.25000	177	0.25000
161	0.25000	145	0.25000	67	0.25000	14	0.25000	2	0.25000

W2DIF Y?

QZ 11

**RETRIEVE?**

**YES**

**EXCLUDE PREVIOUS?:**

四  
五

**SUPPRESS PREV. PRINTED?!**

**YES**

...THERE ARE 19 SPACES IN THE TEMPORARY FILE.

WHERE ARE THE HOLES IN THE  
SPACES IN TEMP. WANT MORE?

YES

YES ACTIVE

DELETE ALL. RETRIEVING

300-11

**SELECT.**

IDENTIFY DOCUMENTS:

**ALL**

TIDEP ID. STAIRS WITH SC.  
COAST PRSENT?

1-2113

WORD	STEM	CONCEPT-WEIGHT PAIRS
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		
19		
20		
21		
22		
23		
24		
25		
26		
27		
28		
29		
30		
31		
32		
33		
34		
35		
36		
37		
38		
39		
40		
41		
42		
43		
44		
45		
46		
47		
48		
49		
50		
51		
52		
53		
54		
55		
56		
57		
58		
59		
60		
61		
62		
63		
64		
65		
66		
67		
68		
69		
70		
71		
72		
73		
74		
75		
76		
77		
78		
79		
80		
81		
82		
83		
84		
85		
86		
87		
88		
89		
90		
91		
92		
93		
94		
95		
96		
97		
98		
99		
100		

>>>>>>>>>>> NO WORDS USED TO FORM THIS QUERY <<<<<<<<<<<<

Figure III-1(b) Dialogue With Experienced User (cont'd)

PRINT QUERY VECTOR?:

= NO

NO. OF HITS= 44.

PRINT TEMP?:

= YES

ACCESSION NUMBER	TEMPORARY IDENT.	CORRELATION [WHEN LAST RETRIEVED]	RANK	BIB. DATA PRINTED/ RET'D LAST QUERY
49	75	0.1187E+00	18	F/T
48	74	0.1535E+00	12	F/T
47	73	0.2571E+00	6	F/T
46	72	0.1461E+00	16	F/T
45	71	0.1088E-01	44	F/T
44	70	0.2575E-01	36	F/T
43	69	0.3144E-01	34	F/T
41	68	0.5843E-01	29	F/T
40	67	0.1489E-01	41	F/T
39	66	0.1465E+00	15	F/T
38	65	0.2806E-01	35	F/T
37	64	0.9174E-01	20	F/T
36	63	0.1842E-01	40	F/T
35	62	0.4263E-01	32	F/T
34	61	0.6924E-01	25	F/T
33	60	0.1334E+00	17	F/T
32	59	0.9310E-01	19	F/T
31	58	0.1727E+00	9	F/T
30	57	0.2343E+00	7	F/T
29	56	0.7417E-01	24	F/T
28	55	0.8832E-01	21	F/T
27	54	0.3707E-01	33	F/T
26	53	0.6106E-01	27	F/T
25	52	0.6209E-01	26	F/T
24	51	0.4350E-01	31	F/T

Figure III-1(b) Dialogue With Experienced User (cont'd)

23	50	0.1164E-01	43	F/T
28	49	0.3289E+00	4	F/T
21	48	0.5463E-01	30	F/T
19	47	0.1327E-01	42	F/T
17	46	0.2374E-01	37	F/T
16	45	0.8028E-01	23	F/T
15	44	0.3575E+00	3	F/T
14	43	0.2090E+00	8	F/T
13	42	0.1000E+01	1	F/T
11	41	0.3706E+00	2	F/T
10	40	0.2006E-01	39	F/T
9	39	0.1687E+00	19	F/T
8	38	0.8222E-01	22	F/T
7	37	0.2050E-01	38	F/T
6	36	0.2890E+00	5	F/T
5	35	0.1629E+00	11	F/T
4	34	0.1469E+00	14	F/T
3	33	0.1511E+00	13	F/T
2	32	0.6029E-01	28	F/T

PRINT BIBLIO.?:

= NO

OPTIONS:

= DOT

INVALID.

OPTIONS:

= DDC

TEMP. DOCS. ONLY, SHORT FORM?:

= NO

IDENTIFY DOCUMENTS:

= All

ACC. NO. 11. ID. 41. CORR. 0.371. RANK 2. RETRIEVED BY LAST QUERY  
 & THE 2 ART OF 4 TEACHING & INFORMATION & SCIENCE .  
 & REES , & ALAN & M.

Figure III-1(b) Dialogue With Experienced User (cont'd)

```

PRINT ABSTRACT?:
= NO
MORE?:
= YES
IDENTIFY DOCUMENTS:
= A15

ACC. NO. 15. ID. 44. CORR. 0.358. RANK 3. RETRIEVED BY LAST QUERY
& TOWARD AN & EDUCATIONAL & BASE FOR THE & INFORMATION & SCIENCES AND
& INFORMATION & ENGINEERING .
& TAYLOR , & ROBERT & S.
PRINT ABSTRACT?:
= NO
MORE?:
= NO
OPTIONS:
= MOD
MORE DOC.--DOC.?:
= NO
TOTALLY REPLACE PRESENT QUERY?:
= NO
QUERY WORD ACTION?:
= NO
DIRECT CON. VECT. ACTION?:
= NO
RETRIEVE?:
= NO
MORE DOC.--DOC.?:
= YES
CONTINUE PREV.?:
= YES

IDENTIFY DOCUMENTS:
= A15
MORE?:
= YES

```

Figure III-1(b) Dialogue With Experienced User (cont'd)

```

= ALL
MMORE? :
= NO DIRECT CON. VECT. ACTION? :
= NO RETRIZVE? :
= YES TEMP. FILE EMPTY TO START.
EXCLUDE PREVIOUS? :
= NO SUPPRESS PREV. PRINTED? :
= NO TEMP ID. STARTS WITH 1.
PRINT PRSNT? :
= YES

```

>>>>>>>>>> NO WORDS USED TO FORM THIS QUERY <<<<<<<<<

**YES**

C&N- CEPT	WEIGHT	C&N- CEPT	WEIGHT	C&N- CEPT	WEIGHT	C&N- CEPT	WEIGHT	C&N- CEPT	WEIGHT	C&N- CEPT	WEIGHT	C&N- CEPT	WEIGHT	C&N- CEPT	WEIGHT
76	0.09091	75	0.29091	62	0.09091	51	0.27273	50	0.09091	50	0.09091	50	0.09091	50	0.09091
46	0.09091	44	0.09091	43	0.09091	34	0.09091	33	0.09091	33	0.09091	33	0.09091	33	0.09091
23	0.18182	17	0.09091	13	0.09091	10	0.09091	7	0.09091	7	0.09091	7	0.09091	7	0.09091
5	0.09091	813	1.10000	748	0.25000	733	0.25000	729	0.25000	729	0.25000	729	0.25000	729	0.25000
727	0.50000	721	0.25000	705	0.25000	654	0.25000	653	0.25000	653	0.25000	653	0.25000	653	0.25000
641	0.25000	620	0.25000	619	0.25000	583	0.25000	583	0.25000	583	0.25000	583	0.25000	583	0.25000
536	0.25000	534	0.25000	533	0.25000	518	0.25000	518	0.25000	518	0.25000	518	0.25000	518	0.25000
461	0.45000	421	3.00000	419	0.25000	358	0.25000	358	0.25000	358	0.25000	358	0.25000	358	0.25000
282	0.25000	227	0.43182	203	0.25000	201	0.25000	201	0.25000	201	0.25000	201	0.25000	201	0.25000
161	0.25000	145	0.25000	67	0.25000	14	0.25000	14	0.25000	14	0.25000	14	0.25000	14	0.25000

NO. OF HITS= 45.

Figure III-1(b) Dialogue With Experienced User (cont'd)

PRINT TEMP?  
= YES

ACCESSION NUMBER	TEMPORARY IDENT.	CORRELATION [WHEN LAST RETRIEVED]	RANK [RETRIEVED]	BIB. DATA PRINTED/ RET'D LAST QUERY
49	45	0.1690E+00	16	F/T
48	44	0.1659E+00	17	F/T
47	43	0.3553E+00	7	F/T
46	42	0.2109E+00	14	F/T
45	41	0.1506E-01	43	F/T
44	40	0.3922E-01	33	F/T
43	39	0.5102E-01	31	F/T
41	38	0.8902E-01	23	F/T
40	37 <sup>a</sup>	0.2239E-01	36	F/T
39	36	0.1882E+00	15	F/T
38	35	0.3525E-01	34	F/T
37	34	0.1181E+00	19	F/T
36	33	0.1596E-01	40	F/T
35	32	0.7768E-01	27	F/T
34	31	0.8342E-01	26	F/T
33	30	0.2444E+00	11	F/T
32	29	0.6581E-01	30	F/T
31	28	0.2875E+00	9	F/T
30	27	0.3712E+00	6	F/T
29	26	0.5007E-01	32	F/T
28	25	0.1222E+00	18	F/T
27	24	0.2030E-01	38	F/T
26	23	0.7181E-01	29	F/T
25	22	0.7286E-01	28	F/T
24	21	0.2382E-01	35	F/T
23	20	0.2120E-01	37	F/T
22	19	0.4623E+00	5	F/T
21	18	0.8622E-01	25	F/T

Figure III-1(b) Dialogue With Experienced User (cont'd)

19	17	0.7266E-02	45	F/T
16	16	0.1260E-01	44	F/T
17	15	0.2009E-01	39	F/T
16	14	0.8837E-01	24	F/T
15	13	0.5133E+00	3	F/T
14	12	0.3500E+00	8	F/T
13	11	0.8803E+00	1	F/T
11	10	0.6480E+00	2	F/T
10	9	0.1538E-01	42	F/T
9	8	0.1029E+00	21	F/T
8	7	0.1165E+00	20	F/T
7	6	0.1572E-01	41	F/T
6	5	0.4664E+00	4	F/T
5	4	0.2629E+00	10	F/T
4	3	0.2261E+00	13	F/T
3	2	0.2438E+00	12	F/T
2	1	0.9905E-01	22	F/T

PRINT BIBLIO.?:  
 = NO  
 OPTIONS:  
 = DDC  
 TEMP. DOCS. ONLY, SHORT FORM?:  
 = NO  
 IDENTIFY DOCUMENTS:  
 = A6  
 ACC. NO. 6.  
 & SOME & REFLECTIONS ON THE & RELATION BETWEEN & INFORMATION & SCIENCE  
 AND & DIGITAL & PROCESSORS .  
 & CHEYDLEUR , & BENJAMIN & F.  
 PRINT ABSTRACT?:  
 = YES

Figure III-1(b) Dialogue With Experienced User (cont'd)

THESE REFLECTIONS ARE CONCERNED WITH THE GROWING IMPORTANCE FOR EDUCATION OF THE REQUIREMENTS ON TOMORROW'S APPLIED RESEARCH PROFESSIONALS IN THE FIELD OF INFORMATION PROCESSING TECHNOLOGY . DETAILED EXEMPLIFICATION OF APPLIED RESEARCH AS A BRIDGE BETWEEN BASIC AND DEVELOPMENTAL ACTIVITIES IS PRESENTED AS NECESSARY FOR SPECIFIC BREAKTHROUGHS IN DIGITAL PROCESSING . FOR DEFINITENESS , INFORMATION SCIENCE IS INTERPRETED AS ENTAILING COMBINATORICS , ANALOGICS , AND FORMAL SYSTEMS . ITS ESSENTIAL ROLE IN MEDIATING TRADEOFFS IN THE DESIGN OF HARDWARE AND EQUIVALENT SOFTWARE PROGRAMMING SYSTEMS IS PRESENTED . THE VALUE OF INFORMATION SCIENCE TO INDUSTRIAL RESEARCH AND DEVELOPMENT MANAGEMENT IS ILLUSTRATED . SEVEN POSITION STATEMENTS DEVELOPED AT A RECENT MEETING OF INFORMATION SCIENTISTS ARE THEN QUOTED TO ILLUSTRATE THEIR NOTION OF THE GROWING MATURATION OF INFORMATION SCIENCE AND THE PROBLEMS OF FOSTERING IT IN EDUCATIONAL AND INSTITUTIONAL ENVIRONMENTS . X & AUTHOR } MORE?

= YES  
IDENTIFY DOCUMENTS:  
= A22

ACC. NO. 22.  
& THE & LIBRARY AS A & PARTNER IN & SCIENTIFIC & CREATIVITY .  
& GARDNER , & JOHN & L.  
PRINT ABSTRACT?:  
= YES

IN THIS AGE OF INFORMATION EXPLOSION THE LIBRARY IS NOT DOING ITS JOB IF IT IS MERELY PASSIVE .. ACQUIRING , LISTING AND DISPLAYING MATERIAL BUT FAILING TO EXPLOIT . THE LIBRARIAN CONTRIBUTES TO SCIENTIFIC CREATIVITY BY DRAWING THE SCIENTISTS ATTENTION TO THE INFORMATION CONTAINED IN HIS COLLECTION , BY SEEKING TO KNOW WHAT THE SCIENTIST NEEDS AND BY PROVIDING TWO PARTICULAR KINDS OF SERVICE .. CURRENT AWARENESS AND RETROSPECTIVE SEARCH FACILITIES . IT PROVIDES THE FORMER BY CIRCULATING TITLES OF PAPERS IN CURRENT JOURNALS , LISTS OF NEW BOOKS AND REPORTS ADDED TO STOCK , NEWS OF SCIENTIFIC CONFERENCES ,

Figure III-1(b) Dialogue With Experienced User (cont'd)

ETC. . THE SECOND SERVICE CAN ONLY BE PROVIDED IF INFORMATION SPECIALISTS & LITERATURE SCIENTISTS ARE EMPLOYED IN THE LIBRARY. THESE MEN NEED TECHNIQUES AND EQUIPMENT DESIGNED TO EXTRACT RELEVANT INFORMATION AS REQUIRED. TRANSLATION SERVICES WILL FREQUENTLY BE NEEDED. THE WRITER, AS MANAGER OF THE & TECHNICAL & INFORMATION & CENTRE AT & GENERAL & PRECISION & AEROSPACE, & LITTLE & FALLS, & N. & J., BRIEFLY DESCRIBES THE ORGANIZATION OF HIS OWN DEPARTMENT IN THE CONCERN. & \$ LSCA ]

MORE?:

- = NO

OPTIONS:

- = END

PRESENT SEARCH QUERY SEQUENCE TERMINATED. A NEW QUERY MAY BE INITIATED AT THIS TIME OR YOU MAY SIGN OFF.

DO YOU WISH TO CONTINUE IN THE SAME MODE?:

- = OPTIONS

ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE OPTIONS):

- = END

PRESENT SEARCH QUERY SEQUENCE TERMINATED. A NEW QUERY MAY BE INITIATED AT THIS TIME OR YOU MAY SIGN OFF.

DO YOU WISH TO CONTINUE IN THE SAME MODE?:

- = END

ANSWER "YES" OR "NO":

- = NO

DO YOU WISH TO TERMINATE USE OF THE SYSTEM?:

- = YES

ON-LINE RETRIEVAL SYSTEM SIGNING OFF.

Figure III-1(b) Dialogue With Experienced User (concluded)

The start of the dialogue illustrates the manner in which the System assists the user when he has trouble; even an experienced user will occasionally rely on this feature. In this case, the user indicates that he does not want normal operation, and refuses an explanation of the modes that are available to him. But when he is asked to identify flags to be turned on, he realizes that he has forgotten how the flags are used. When he enters "WHAT?" to the question about flags to be set, he is given a second opportunity to see the list of modes, which he uses.

The user elects to set flags 1, 3, and 4. Flag 1 selects the terse form of System messages to the user; flag 3 enables the user to obtain a complete analysis of his query before performing a retrieval, and flag 4 permits the user to print the ranking table after a retrieval.

Once the user has selected mode flags, he must choose an option. He enters "HELP" to see the list of options; but he has selected terse dialogue, and therefore does not receive a complete explanation of their purposes. To get this explanation, he uses option "CHG" to cancel his selection of terse dialogue, gets the explanation of all options, restores his previous flag selections, and continues.

The query the user enters is identical to the previous user's query, as discussed above, including the misspelling of "broadest" as "boradest". Realizing his typing error, the user corrects his mistake by entering the word, typed correctly. It is not necessary for him to delete the misspelled version, since words whose stems are not in the stem-concept dictionary are not retained in the query. In order to observe the System's processing of his query, the user prints the query concept vector and present table. His query contained ten content-stem-producing words. "Information" appeared three times, accounting for the weight of 3.0 assigned to concept number 421 in the query vector. "Processing" occurred twice, giving a weight of 2.0 to concept number 648.

The retrieval produces the same result obtained in the earlier dialogue, since the same query was used. The user moves directly into option "DOC" to print the abstract of A13, the highest-ranked document retrieved. This document is very similar to his query; therefore, he decides to use document-document searching to find another document similar to A13. It is expected that most query sequences will be conducted in this way, since document-document searching is a rapid way to construct a complex query vector.

In the results of the document-document search, A13 is of course ranked highest, since it correlates perfectly with itself. The user prints bibliographic data for A11 and A15, which are ranked 2 and 3. Satisfied that these documents are of interest, he decides to perform document-document searching based on both of these documents, and A13. Note that the number of concepts in the query concept vector has grown from 7, for the initial query, to 34, for the first document-document search, to 50, for the present document-document search based on three documents. In this manner, the user is building an increasingly complex query vector, that describes more and more precisely the concept vector of the document he seeks. In this case, the experienced user, by using the multiple-document facility of document-document searching, has developed a much more sophisticated concept vector than the beginning user, who merely modified his previous query by adding a few words.

At this point, the user decides to inspect the abstracts of the highest-ranked documents in the latest retrieval that have not yet been printed, namely A6 and A22. At this point, he has read abstracts of five documents that are apparently of interest, so he signs off the System. Presumably he would obtain hard-copy (or microimage) versions of the documents. If these documents did not fulfill his need, he might browse further by initiating another query sequence.

## III. 2 FILES ACCESSED BY THE ON-LINE SYSTEM

This section discusses in some detail each of the files accessed by the On-Line System. Figure III-2 names these files and describes their contents. The word "file" is used here to refer to a collection of data that is logically distinct, rather than to indicate any sort of programming method. Thus, some of these "files" are actually GECOS III quick-access files, as shown in Figure III-2(a), while others might be called "pseudo-files", since they are stored as part of the dialogue processor (Figure III-2(b)).

### III. 2.1 Document File--DATA1

The document file, DATA1, consists of fifty abstracts, with bibliographic data, furnished by RADC for purposes of testing the dialogue processor. Figure III-3 shows the beginning and end of DATA1.

### III. 2.2 File Words

This file contains five types of information: common words and suffixes of one, two, three and four characters. The file is indexed by line number, with the first digit of the line number indicating the type of information stored on that line. This digit is 0 for common words and 1, 2, 3, or 4 for stems of the corresponding length. The next three digits are for sequencing only. Figure III-4 is a listing of WORDS.

### III. 2.3 Dictionary File

The stem-concept dictionary consists of a list of stems and concept vectors. With each stem is stored a three-component concept vector. The first component is a unique concept number with a weight of unity, and the second and third components are null. This dictionary is used to map a query or document into a concept vector.

File Name	Contents	When Printed
DATA1	For each document: <ul style="list-style-type: none"> <li>• Title</li> <li>• Author</li> <li>• Abstract</li> </ul>	Selective printing for retrieved documents.
WORDS	Common words and suffixes for stem analysis.	Not printed.
DICTNRY	Dictionary--for each content stem, the stem and its corresponding concept-weight pairs.	Not printed.
CONCEPTS	One concept vector for each document in DATA1.	Printed by option "CON".
MESSAGES	Catalog of user messages that can be transmitted by the On-Line System.	Printed as required.
OFFLINE	Abstracts to be printed off-line.	Not printed by the Dialogue Processor. (see III. 2. 6)

Figure III-2(a) Files Accessed by the  
On-Line System: GECOS III Files

File Name	Contents	When Printed
QUERY VECTOR	Concept-weight pairs for the present query.	When requested under option MOD, or under mode number 2.
PRESENT QUERY	For each word in the present query: <ul style="list-style-type: none"> <li>• the word</li> <li>• its stem</li> <li>• the concept-weight pairs for the stem</li> </ul>	When requested under option MOD, or under mode number 3.
TEMPORARY FILE	For each document retrieved: <ul style="list-style-type: none"> <li>• accession number</li> <li>• temporary identification number</li> <li>• correlation when last retrieved</li> <li>• rank when last retrieved</li> <li>• flag indicating whether bibliographic data has been printed during this query sequence</li> <li>• flag indicating whether there was a "hit" during the last retrieval in this sequence</li> </ul>	When requested under option DOC or under mode number 4.

Figure III-2(b) Files Accessed by the  
On-Line System: Core-Resident Files

\*\*TITLE  
 & IDENTIFYING AND & LOCATING & STANDARDS .  
 \*\*AUTHOR  
 & ASTALL, & K.  
 \*\*ABSTRACT  
 & INITIAL MICROFICHE FOR EACH REPORT 1 , & 4 1 IMAGE AREA REQUIREMENTS  
 LIBRARIANS HAVE TO DEAL WITH THREE MAIN TYPES OF STANDARD--MATERIAL  
 SPECIFICATIONS , STANDARDS OF QUALITY AND CODES OF PRACTICE . THEY CAN  
 COVER PRACTICALLY ANY SUBJECT AND CAN BE PRODUCED BY INTERNATIONAL OR  
 NATIONAL ORGANIZATIONS , TRADE ASSOCIATIONS OR INDIVIDUAL FIRMS .  
 ONE OF THE GREATEST DIFFICULTIES ENCOUNTERED IN DEALING WITH STANDARDS  
 IS IDENTIFICATION , AS A PROLIFIC NUMBER OF SYMBOLS ARE USED BY  
 ORIGINATORS OF STANDARDS . & \$ LSCA 1  
 \*\*END

\*\*TITLE  
 & FEDERAL & MICROFICHE & STANDARDS .  
 \*\*AUTHOR  
 & FEDERAL & COUNCIL FOR & SCIENCE AND & TECHNOLOGY .  
 \*\*ABSTRACT  
 THE GENERAL CHARACTERISTICS OF THE \$ COSATI & COMMITTEE ON  
 & SCIENTIFIC AND & TECHNICAL & INFORMATION 1 MICROFICHE ARE GIVEN .  
 THE SPECIFICATIONS HAVE SECTIONS CONCERNING & 1 1 FILMING REQUIREMENTS  
 OF THE MICROIMAGE AREA , & 2 1 REQUIREMENTS IN THE TITLE AREA & INITIAL  
 MICROFICHE FOR EACH REPORT 1 , & 3 1 DOCUMENT IMAGE AREA REQUIREMENTS  
 TRAILER MICROFICHE 1 , AND & 5 1 QUALITY . & \$ ARDK 1

INFORMATION & RETRIEVAL OR & THE & BIRTH OF A & MYTH .

ILL EFFECTS OF THE  
 YEARS AND  
 STATES

Figure III-3 DATA1 (cont'd)

\*\*END  
 \*\*TITLE  
 & DESCRIPTION  
 \*\*AUTHOR  
 & U. S. & ATOMIC  
 \*\*ABSTRACT  
 DETAILED DIRECTIONS ARE  
 THE PREPARATION OF DESCRIPTION  
 PERMITS COMPUTER MANIPULATION  
 PROVIDE A VARIETY OF SERVICES  
 \*\*END  
 \*\*TITLE  
 \*\*AUTHOR  
 & ADAMS, & SCOTT .  
 \*\*ABSTRACT  
 OPERATIONAL EXPERIENCE DURING THE FIRST YEAR OF MEDLARS IS  
 DESCRIBED IN RELATION TO THE & LIBRARIES PRIMARY OBJECTIVES OF INFORMATION  
 PUBLICATION, RECURRING BIBLIOGRAPHIES, AND DEMAND SEARCH SERVICES  
 PROGRESS TOWARD THE IMPLEMENTATION OF TWO SECONDARY OBJECTIVES--THE  
 INPUT OF CATALOGING COPY AND THE DECENTRALIZATION OF THE SYSTEM--IS  
 ALSO REPORTED & MEDLARS HAS BEEN INTENSIVELY TESTED DURING ITS FIRST  
 OPERATIONAL YEAR, ITS STRENGTHS AND WEAKNESSES HAVE BEEN IDENTIFIED,  
 AND IT IS NOW PREPARED TO DEVELOP ITS SERVICE POTENTIAL. % \$ AUTHOR ]  
 \*\*END

Figure III-3 DATA! (concluded)

0001 ABOUT	0049 DOING	0097 IN
0002 ABOVE	0050 DONE	0098 INSOFAR
0003 ACROSS	0051 DO	0099 INSTEAD
0004 AFTER	0052 DOWN	0100 INTO
0005 AGAINST	0053 DURING	0101 INWARD
0006 ALL	0054 EACH	0102 I
0007 ALMOST	0055 EITHER	0103 IS
0008 ALONE	0056 ELSE	0104 IT
0009 ALONG	0057 ELSEWHERE	0105 ITSELF
0010 ALSO	0058 ENOUGH	0106 ITS
0011 ALTHOUGH	0059 ETC	0107 JUST
0012 ALWAYS	0060 EVEN	0108 KEEP
0013 AMONG	0061 EVER	0110 LEAST
0014 AM	0062 EVERYONE	0111 LESS
0015 AND	0063 EVERY	0112 LEST
0016 ANOTHER	0064 EVERYTHING	0113 MANY
0017 AN	0065 EVERYWHERE	0114 MAY
0018 ANYBODY	0066 EXCEPT	0115 ME
0019 ANYONE	0067 FEW	0116 MIGHT
0020 ANY	0068 FOR	0117 MINE
0021 ANYTHING	0069 FORTH	0118 MOREOVER
0022 ANYWHERE	0070 FROM	0119 MORE
0023 APART	0071 FURTHERMORE	0120 MOST
0024 ARE	0072 GET	0121 MUCH
0025 AROUND	0073 GETS	0122 MUST
0026 A	0074 GOT	0123 MY
0027 ASIDE	0075 HAD	0124 MYSELF
0028 AS	0076 HARDLY	0125 NEITHER
0029 AT	0077 HAS	0126 NEVERTHELESS
0030 AWAY	0078 HAVE	0127 NEXT
0031 AWFULLY	0079 HAVING	0128 NOBODY
0032 BECAUSE	0080 HENCE	0129 NONE
0033 BEEN	0081 HEREIN	0130 NOR
0034 BEFORE	0082 HERE	0131 OR
0035 BEHIND	0083 HER	0132 NO
0036 BEING	0084 HERSELF	0133 NOTHING
0037 BELOW	0085 HE	0134 NOT
0038 BE	0086 HIM	0135 NOWHERE
0039 BETWEEN	0087 HIMSELF	0136 OF
0040 BEYOND	0088 HIS	0137 OH
0041 BOTH	0089 HITHER	0138 ONE
0042 BUT	0090 HOWBEIT	0139 ONES
0043 BY	0091 HOWEVER	0140 ONLY
0044 CANNOT	0092 HOW	0141 ON
0045 CAN	0093 IF	0142 OUT
0046 COULD	0094 INASMUCH	0143 OTHER
0047 DID	0095 INDEED	0144 OTHERS
0048 DOES	0096 INNER	0145 OTHERWISE

Figure III-4 WORDS Listing  
(continued)

0146	UGHT	0194	THOUGH	1002	E	3028	MEN
0147	OUR	0195	THROUGHOUT	1003	S	3029	MAN
0148	OURSELVES	0196	THUS	1004	Y	3030	NØT
0149	OURS	0197	TØGETHER	2001	'S	3031	ØDE
0150	OUTSIDE	0198	TØØ	2002	AL	3032	ØSE
0151	ØVER	0199	TØ	2003	AN	3033	ØUS
0152	ØWN	0200	TØWARD	2004	AR	3034	PLY
0153	PER	0201	TWØ	2005	CY	3035	STY
0154	PLEASE	0202	UNDERNEATH	2006	ED	3036	TAL
0155	PLUS	0203	UNDER	2007	ED	3037	TER
0156	QUITE	0204	UNLESS	2008	EN	3038	TIC
0157	RATHER	0205	UNTIL	2009	ER	3039	TLE
0158	REALLY	0206	UNTØ	2010	ET	3040	ULE
0159	RIGHT	0207	UPØN	2011	IC	3041	URE
0160	SELF	0208	UP	2012	LY	3042	VAR
0161	SELVES	0209	US	2013	ØN	3043	WAY
0162	SEVERAL	0210	VERY	2014	ØR	4001	ABLE
0163	SHALL	0211	WAS	2015	ØU	4002	ANCE
0164	SHE	0212	WELL	2016	RY	4003	CANT
0165	SHØULD	0213	WERE	2017	S'	4004	CIDE
0166	SINCE	0214	WE	2018	TH	4005	DUCE
0167	SIX	0215	WHATEVER	3001	AGE	4006	ENCE
0168	SØMEBØDY	0216	WHAT	3002	ANT	4007	EVER
0169	SØME	0217	WHENCE	3003	ARY	4008	HAND
0170	SØMETHING	0218	WHENEVER	3004	ATE	4009	IENT
0171	SØMETIMES	0219	WHEN	3005	BAR	4010	ITIE
0172	SØMEWHAT	0220	WHERE	3006	CAN	4011	LENT
0173	SØ	0221	WHEREVER	3007	DER	4012	LERT
0174	STILL	0222	WHETHER	3008	EED	4013	LESS
0175	SUCH	0223	WHICH	3009	ENT	4014	MATE
0176	TEN	0224	WHILE	3010	EST	4015	MENT
0177	THAN	0225	WHØM	3011	ETH	4016	MITY
0178	THAT	0226	WHØ	3012	FUL	4017	NESS
0179	THEIR	0227	WHØSE	3013	GEN	4018	PEAR
0180	THEIRS	0228	WHY	3014	IAL	4019	SERT
0181	THEM	0229	WILL	3015	IAN	4020	SERT
0182	THEMSELVES	0230	WITHIN	3016	IED	4021	SHIP
0183	THENCE	0231	WITHOUT	3017	IES	4022	SING
0184	THEN	0232	WITH	3018	ING	4023	SØRB
0185	THEREBY	0233	WØULD	3019	ISH	4024	TEEN
0186	THEREFØRE	0234	YES	3020	ISM	4025	THER
0187	THERE	0235	YET	3021	IØN	4026	TIAL
0188	THE	0236	YØUR	3022	IST	4027	TIØN
0189	THESE	0237	YØURSELF	3023	ITY	4028	TIZE
0190	THEY	0238	YØURSELVES	3024	IVE	4029	TURB
0191	THIS	0239	YØURS	3025	IZE	4030	VICE
0192	UPWARD	0240	YØU	3026	LAY	4031	WISE
0193	THØSE	1001	'	3027	LEL		

Figure III-4 WORDS Listing  
(concluded)

This file is generated by program DICGEN, which reads from the test file of 50 documents. It reads a record and skips the rest of the record if the record is title, author, end or corporate information. If the record is the beginning of an abstract, the abstract is searched for stems. Words which do not begin with an alphabetic character are excluded from consideration. The stem analysis routine is employed, so that common words are rejected. The program generates a table of up to 5000 stems.

The program first calls for the maximum number of dictionary entries. It then processes abstracts from the DATA1 file until the specified number of unique stems has been found, or the end of the file has been reached. A PLUCK delimiter parameter of two is used, so words containing hyphens are not split. It is required that the first character of any stem be alphabetic, and stems not meeting this requirement are rejected. The dictionary is alphabetized.

The program PLUCKs from DATA1 until the beginning of an abstract is found. It then PLUCKs and STEMs, rejecting common words through STEM and stems that do not start with a letter of the alphabet directly. A stem that meets these requirements is checked against the dictionary, and if it is not already in the dictionary, it is added in the correct order.

Programs that generate little or no Teletype output can go dead owing to computer failure, like any other programs. But if there is no expected output, the user cannot detect the error. To avoid this, a bell is rung at the Teletype when a stem is entered into the dictionary.

The program was first written to produce only a list of stems. Once this had been produced, then the program was modified to generate the stem-concept dictionary. Figure III-5 shows the 900 stems presently included; the program can generate up to 5000 stems. Figure III-6 contains the flowchart and Figure III-7 is a listing of the first twenty dictionary entries. A listing of the program itself appears in Section VI.

AO	ABILIT	ABLE	ABREAST	ABSENC
ABSTRACT	ACADEM	ACCEPT	ACCES	ACCLAIM
ACCOUNT	ACCREDITA	ACHIEV	ACQUIR	ACQUI SI
ACTIV	ACTIVIT	ACTUAL	ADAPT	ADDED
ADDIT	ADDS	ADEQU	ADMINISTRA	ADMINISTRAT
ADVANC	ADVANT	AEROSPAC	AFFAIR	AFFECT
AGE	AGENC	AGREE	AGREED	AID
AIDS	AIMED	AIMS	ALICE	ALLEG
ALLOW	AMERI	ANALOG	ANALYSI	ANALYST
ANALYZ	ANNUAL	APPEAR	APPLI	APPLICA
APPROACH	APT	ARCHIV	AREA	AREAS
AGENT	ARISE	ARRANG	ARRIV	ARTICL
ARTICULA	ARTIS	ASKED	ASPECT	ASSES
ASSIGN	ASSIMIL	ASSIST	ASSOCIA	ATTAIN
ATTEMPT	ATTEN	ATTEND	ATTITUD	AUTHOR
AUTHORO	AUTOMA	AVAIL	AVAILABIL	AWARE
BACHELORO	BACK	BACKGROUND	BACKLOG	BALANC
BAR	BASED	BASIC	BASIS	BEARING
BEARS	BECOM	BEGIN	BEGUN	BENEFIT
BETTER	BIBLIOGRAPH	BILOG	BLOOMQU	BLUR
BODY	BOOK	BOOKMOBIL	BOOKS	BOUND
BRANCH	BREAKTHROUGH	BRIDGE	BRIEF	BRIEFL
BRIEFLI	BROWS	BRYANO	BUILT	BURDEN
CANAD	CAPABILIT	CARD	CARDS	CARRI
CARRY	CATALOG	CATALOGU	CAUSE	CENTER
CENTR	CENTRALIZ	CERTAIN	CHALLENG	CHANG
CHARACTERIS	CHECK	CHEMIC	CHIEF	CHURCH
CIRCULA	CIRCULAT	CITAT	CITY	CLAIM
CLAS	CLASSIC	CLASSIF	CLASSIFICA	CLB
CLIENT	CLIENTEL	CO-OPER	CO-OPERA	CO-OPERAT
CO-ORDINAT	CODES	COLLEC	COLLECT	COLLEG
COMBINATOR	COMED	COMING	COMMENC	COMMIT
COMMONO	COMMON	COMMUNICA	COMPET	COMPETIT
COMPIL	COMPILA	COMPL	COMPLEX	COMPON
COMPRI	COMPU	CONCENTR	CONCER	CONCERN
CONFER	CONSI	CONSIDERA	CONSIST	CONST
CONSTITUT	CONSULT	CONTAIN	CONTENT	CONTINU
CONTRACT	CONTRIBU	CONTRIBUT	CONTROL	CONVER
COORDIN	COORDINAT	COPE	COPYING	CORE
CORRELA	COSATI	COST	COURS	COVER
CREAT	CREATIV	CURRENT	CURRICULUM	DARLING
DATE	DAYS	DEAL	DEALING	DEARTH
DECENTR	DECID	DECRE	DEFICIENC	DEFIN
DEFINI	DEFINIT	DEGRE	DELAY	DELIMITA
DELINEA	DEPART	DEPLO	DEPOSIT	DESCRIB
DESCRIP	DESIGN	DESIGNAT	DESIRABIL	DETAIL
DETERMIN	DEVELOP	DEVIC	DIALOGU	DICTI

Figure III-5 (cont'd)

Stem Dictionary

DIFFER	DIFFICULT	DIGIT	DIMIN	DIRECT
DIRECTO	DISAGR	DISCHARG	DISCIPLIN	DISCOV
DISCU	DISCUS	DISPLA	DISPUT	DOCUM
DOCUMENT	DOCUMENTA	DOLLAR	DOUBL	DRAWING
DRAWN	DUE	DUPLIC	DUPLICAT	E
EARLY	EASIER	ECONOM	EDITED	EDITING
EDUCA	EFFECT	EFFIC	EFFORT	ELECTR
ELEMENT	EMERG	EMINENC	EMPHAS	EMPHASI
EMPLØ	ENCOUN	ENCOUR	ENCOURAG	END
ENDS	ENERG	ENGINE	ENTAIL	ENTER
ENTIR	ENVIR	ENVIRON	EQUIP	EQUIVA
ERA	ERROR	ESSEN	ESTABL	EVENT
EVENTU	EVIDENC	EXAMPL	EXCEL	EXCEP
EXEMPLIFICA	EXIST	EXPANS	EXPECT	EXPEN
EXPERI	EXPERIMENTA	EXPLICA	EXPLO	EXPLOIT
EXPLORA	EXPRE	EXTEN	EXTENT	EXTERN
EXTRACT	F	FACILIT	FACTS	FACULT
FAILING	FAILUR	FALLS	FAMILI	FAVOUR
FEATUR	FEDER	FEEL	FIELD	FIL
FILED	FILMING	FINAL	FINANC	FIND
FINDING	FIRMS	FIRST	FISCAL	FIVE
FLOW	FOLLOW	FORCE	FORM	FORMAL
FORMED	FORMER	FORMS	FORMUL	FOSTER
FOUR	FREQU	FRUIT	FULLY	FUNCT
FUNDING	FURTH	FUTUR	G	GAINED
GAVRILØV	GENER	GEOGRAPH	GEORGIA	GETTING
GIVEN	GOALS	GOOD	GOVERN	GRADU
GRAVE	GREAT	GROUP	GROWING	GROWTH
GUEST	GUIDANC	GUIDE	GUIDELIN	HABIT
HALTED	HAND	HARDER	HARDW	HAROLD
HELD	HELP	HETEROGEN	HIGHER	HIGHL
HOMES	HOMOGEN	HOMOMORPH	HOPE	HOSPI
IDENTIFICA	IGNØR	II	III	IL
ILLUSTR	ILLUSTRAT	IMAGE	IMPING	IMPLE
IMPLEMENTA	IMPORT	IMPOS	IMPROV	INADEQU
INCLUD	INCOMPET	INCREA	INDEX	INDISPEN
INDIVIDU	INDIVIDUALØ	INDIVIDUALI	INDUSTR	INFORM
INFORMA	INGENU	INHIBIT	INITI	INSERT
INSTANC	INSTITU	INSTRU	INSTRUC	INTEGR
INTEGRA	INTER	INTERDISCIPL	INTERFAC	INTERFIL
INTERMEDI	INTERNA	INTERPR	INTERPRETA	INTRODUC
INVENT	INVENTØ	INVESTIGA	INVOLV	ISOLAT
ISOMORPH	ISSUE	ITEM	ITEMS	IV
J	JØB	JØURN	JUDGING	JUNIOR
JUSTIF	KEPT	KEY	KINDS	KNOW
KNOWLEDG	KNOWN	L	LABEL	LACK
LACKS	LARGE	LARGER	LATER	LATTER
LAW	LC	LEANS	LECTUR	LEGAL

Figure III-5 (cont'd)

Stem Dictionary

LEI SUR	LENGTH	LETTER	LEVEL	LIAIS
LIBER	LIBRA	LIBRAR	LIBRARIANSO	LIBRARIANSHI
LIBRARYO	LIMIT	LIMITA	LINES	LINGUIS
LINK	LISTED	LISTING	LISTS	LITERAT
LITTL	LIVEL	LOAD	LOCAL	LOCAT
LOGIC	LØNGER	LØUIS	LØWER	LSCA
MACHIN	MAIN	MAINTAIN	MAJØR	MAKE
MAKING	MAN	MANAG	MANNER	MARKET
MASTER	MASTERO	MATER	MATHEMA	MATTER
MATURA	MAXIMUM	MEANING	MEANS	MEASUR
MECHAN	MEDIA	MEDIAT	MEDIC	MEDICIN
MEDLAR	MEETING	MEMBER	MEN	MEREL
MERGE	METHØD	METHØDØLOG	MICRØFICH	MICRØFILM
MICRØFORM	MICRØIM	MINØR	MISCØNCEP	MISPLAC
MØBIL	MØDIF	MØNEY	MØVING	MYTH
N	NAME	NAMEL	NARROW	NATION
NATUR	NECES	NEED	NEEDED	NEEDING
NEEDS	NETWORK	NEW	NEWNE	NEWS
NØN-ACADEM	NØN-SUBJECT	NOTED	NØTION	NØW
NUMBER	NUMER	ØBJECT	ØBLIGA	ØBTAIN
ØCCUR	ØFFIC	ØFTEN	ØNE-SEME	ØPEN
ØPERA	ØPERAT	ØRDER	ØRGANIZ	ØRGANIZA
ØRIGIN	ØRIGINAT	ØUT	ØUTLET	ØUTLIN
ØVERLAP	PAGE	PAGING	PAPER	PAPERBACK
PARAME	PARENT	PARØCH	PART	PARTICIPA
PARTICUL	PARTS	PASSIV	PAST	PATENT
PATRØNØ	PATTERN	PEØPL	PERFORM	PERIODIC
PERSON	PERSONNEL	PERTAIN	PHARMA	PHYSIC
PLACE	PLACED	PLAN	PLAY	PLAYED
PØINT	PØLIC	PØLIT	PØPUL	PØSES
PØSIT	PØSSE	PØSSIBL	PØSSIBL	PØSTS
PØTEN	PØWER	PRACT	PRACTIC	PRECI
PREDØMIN	PREPAR	PRESENT	PRESENTA	PREVENT
PRIMA	PRIMAR	PRINCIP	PRINCIPL	PRINT
PRØBAB	PRØBLEM	PRØCE	PRØCES	PRØCUR
PRØDUC	PRØDUCT	PRØFE	PRØFES	PRØFESSION
PRØGRAM	PRØJECT	PRØLIF	PRØLIFERA	PRØPERT
PRØPØS	PRØSPECT	PRØVI	PRØVID	PRØXI
PSYCHØLOG	PUBLIC	PUBLICA	PUBLICATIØN-	PUBLIS
FURPØ	PURPØS	QUALIF	QUALIFICA	QUALIT
QUANTITAT	QUEST	QUESTIØNNAIR	QUØTA	QUØTED
RADIO	RANGING	RE-PRØGRAM	READ	READING
REAL	REALIZ	REASON	RECATALØGU	RECEIV
RECENT	RECLASSIFICA	RECØGNIZ	RECOMMENDA	RECORD
RECUR	REDUC	REFER	REFLEC	REGARD
REGION	REINFØRC	RELAT	RELEV	REMAIN
REMOV	REØRGANIZA	REPACKAG	REPLI	REPLY
REPORT	REPRESENTAT	REPRØDUC	REQUIR	REQUIRE

Figure III-5 (cont'd)

Stem Dictionary

RESEARCH	RESEARC	RESERV	RESOURC	RESPON
RESPONSI BL	REST	RESULT	RETRIEV	RETROSPECT
REVIEW	REVOLU	RI SE	ROLE	ROLES
ROOT	S	SAME	SANCT	SCALE
SCHDM	SCHOL	SCHOOL	SCIENC	SCIENCE--MAT
SCIENT	SCIENTIF	SCIENTISTO	SEARCH	SECOND
SECTI	SEEKING	SEEN	SELEC	SELECT
SENT	SEQUENC	SERIOU	SERVE	SERVIC
SERVING	SET	SEVEN	SHELV	SHIFT
SHOP	SHOWING	SITUA	SKIL	SMAL
SO-CAL	SOCIET	SOCIOLOG	SOFTW	SOON
SOURC	SOVIET	SPECI	SPECIAL	SPECIALIZA
SPECIF	SPECIFICA	SPENT	STAF	STAND
STANDARD	STANDARD--MA	STARS	STATE	STATI
STATIS	STEAD	STEM	STEMS	STEPS
STOCK	STOREH	STORING	STRATEG	STRONG
STRUCT	STUDENT	STUDI	STUDY	SUBJECT
SUBSEQU	SUGGES	SUGGEST	SUM	SUPERF
SUPPL	SUPPORT	SURPRI	SURVE	SWERV
SYMBOL	SYNTHESI	SYSTEM	SYSTEMO	SYSTEMATIZ
TAKE	TAKEN	TAPES	TASKS	TEACH
TEAM	TEAMS	TECHN	TECHNIQU	TECHNIQUES--
TECHNOLOG	TELEVI	TENTAT	TERMS	TEXTBOOK
THEORE	THINK	THREE	THROUGH	THURSTONEO
TIME	TIMING	TITLE	TOMORROWO	TOOLS
TOPIC	TOTAL	TOWARD	TRADE	TRADEOF
TRADI	TRAIL	TRAIN	TRANSLA	TRAVEL
TREND	TRIVI	TRYING	TWICE	TYPES
U	UCLA	UNCATALOG	UNDECID	UNDERGRADU
UNFAVOUR	UNINFORMAT	UNION	UNIQUE	UNITED
UNIVER	UNIVERSIT	UNNAM	UPHEAV	USA
USE	USED	USER	USERS	UTILIZ
V	VALUE	VARIET	VARIQU	VI
VIABL	VIEWPOINT	VIGOR	VOLUN	WARREN
WATERED-DOWN	WAY	WAYS	WESTERN	WIDEL
WIDER	WORD	WORK	WORKER	WORKING
WORTH	WRITER	X	XROR	YEARS

Figure III-5 (concluded)

Stem Dictionary

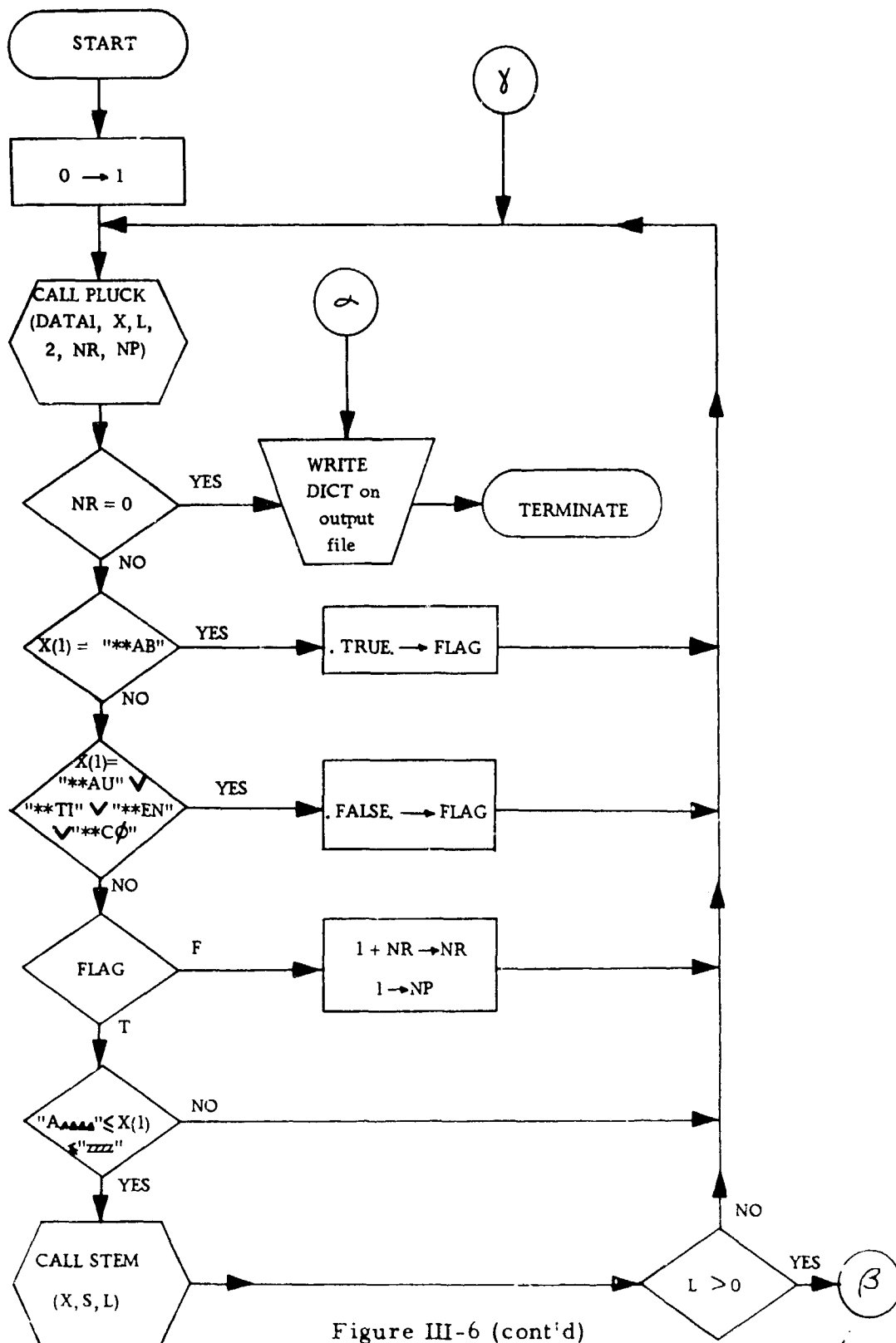


Figure III-6 (cont'd)  
DICTIONARY GENERATOR

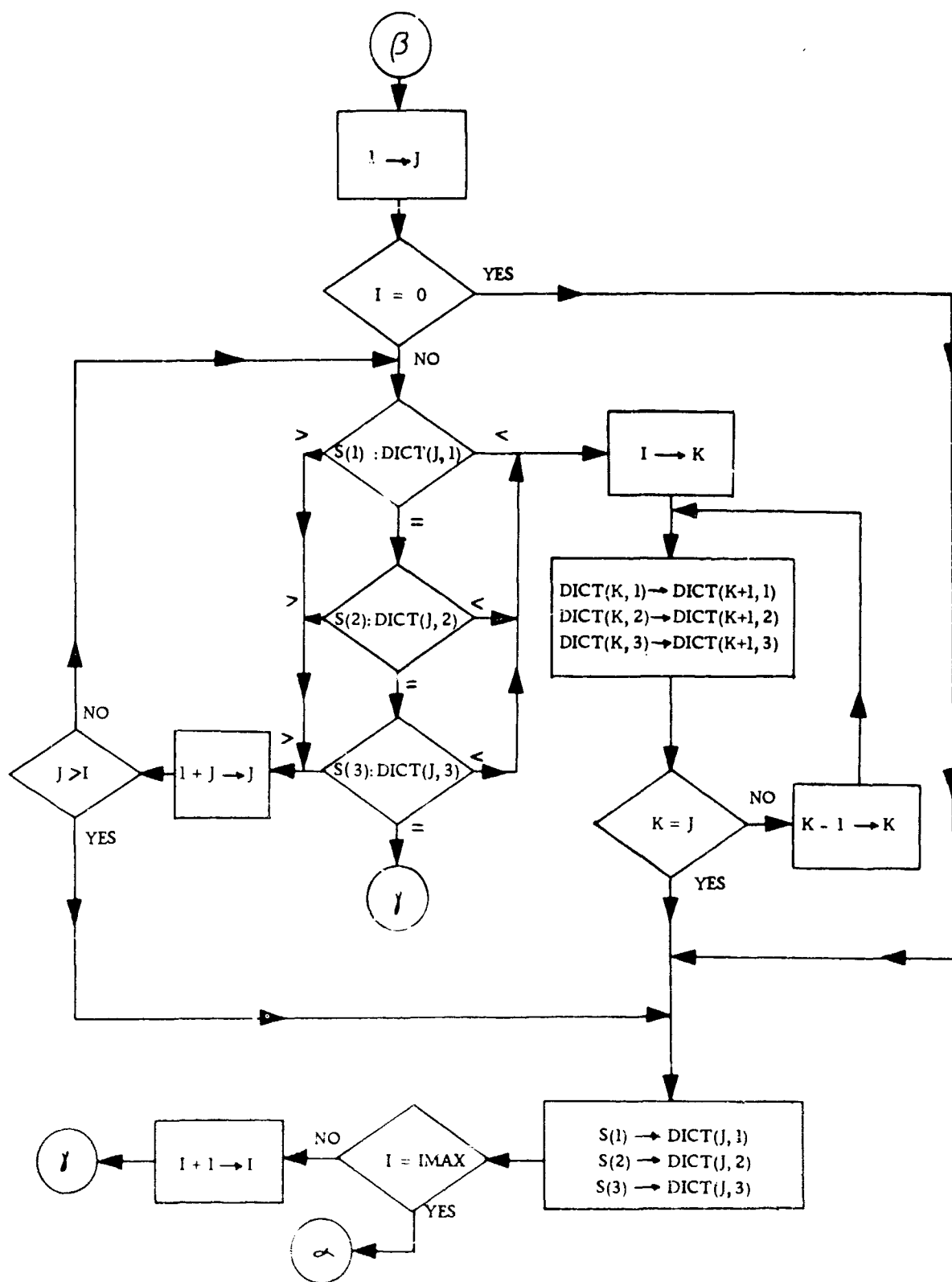


Figure III-6 (concluded)  
DICTIONARY GENERATOR

AO	1	1.00000	0	0.	0	0.
ABILIT	2	1.00000	0	0.	0	0.
ABLE	3	1.00000	0	0.	0	0.
ABREAST	4	1.00000	0	0.	0	0.
ABSENC	5	1.00000	0	0.	0	0.
ABSTRACT	6	1.00000	0	0.	0	0.
ACADEM	7	1.00000	0	0.	0	0.
ACCEPT	8	1.00000	0	0.	0	0.
ACCES	9	1.00000	0	0.	0	0.
ACCLAIM	10	1.00000	0	0.	0	0.
ACCØUNT	11	1.00000	0	0.	0	0.
ACCREDITA	12	1.00000	0	0.	0	0.
ACHIEV	13	1.00000	0	0.	0	0.
ACQUIR	14	1.00000	0	0.	0	0.
ACQUI SI	15	1.00000	0	0.	0	0.
ACTIV	16	1.00000	0	0.	0	0.
ACTIVIT	17	1.00000	0	0.	0	0.
ACTUAL	18	1.00000	0	0.	0	0.
ADAPT	19	1.00000	0	0.	0	0.
ADDED	20	1.00000	0	0.	0	0.
	21	1.00000	0	0.	0	0.

Figure III-7

Dictionary

It should be remembered that there are several parameters of the program that can be varied at will, and it is through inspection of the results such as those in this report that the parameters can be refined.

Adjustable are:

- Common words in the file WORDS, by adding or deleting.
- Specific stems in the file WORDS, by adding and deleting.
- Treatment of the hyphen as a delimiter or not, by selecting 3 or 2 as a parameter of PLUCK.
- Varying the minimum stem length generated by removal (exclusive of treatment of double consonants and "i" before "ly"), by varying a single number in STEM.
- Varying the number of passes allowed, by varying a single number in STEM.

Figure III-5 gives considerable insight into the operation of the stem analyzer. Inspection of the Figure reveals that the stem analyzer has done a good job; there are very few adjacent stems that are forms of the same word. An investigation of the number of artificial homographs created by stem analysis would require a side-by-side comparison of words and the stems they generate; thus, no evaluation of this aspect of stem analyzer performance can be based on Figure III-5 alone.

An important use of Figure III-5 is in determination of the settings of the various parameters of the stem analysis process. For example, it appears that the minimum stem length should be set at four characters, rather than five, which is the present setting. This would reduce the size of the stem dictionary. For example, the words "need", "needed", "needing", and "needs" would all be mapped into "need" if the minimum stem length were reduced to four characters. Other examples of four-letter stems that would each be produced from several entries presently in the stem dictionary are "stud", "form", and "item".

It is not clear from Figure III-5 how the hyphen should be treated during stem analysis--as an alphabetic character or as a word delimiter. The indication is that dictionary size would be reduced by this change, and that few artificial homographs would be created.

#### III. 2. 4     File CONCEPTS

File CONCEPTS contains one concept vector for each document in the collection. This File is generated by program CONGRA.

Program CONGRA first reads the dictionary file, DICTNRY. It then processes the data base (DATA1), finding the stem for each word, and looking up the stem in the dictionary. The weight of each stem in the document is the number of its occurrences, normalized so that the largest weight in each document is unity. Up to fifty components of the concept vector are entered for each document.

Since statistical filtering and document clustering are not performed by the presently operating experimental prototype of the System, some doubt existed whether this simple concept vector file would be sufficient for testing the dialogue processor. Figure III-8 shows both the number of lines of text processed as the program proceeded through the document collection, and the number of components in the concept vectors for fifty documents, and illustrates that the present file will be sufficient for testing purposes. Figure III-9 is a listing of part of CONCEPTS, and Figure III-10 is a flowchart of CONGRA.

#### III. 2. 5     File MESSAGES

File MESSAGES contains the list of System-user messages, with five-digit line numbers. For each message, the terse form is stored in the file before the verbose form. Within either form, the lines of multiple-line messages appear in the order in which they are printed. The line numbers identify the messages as follows:

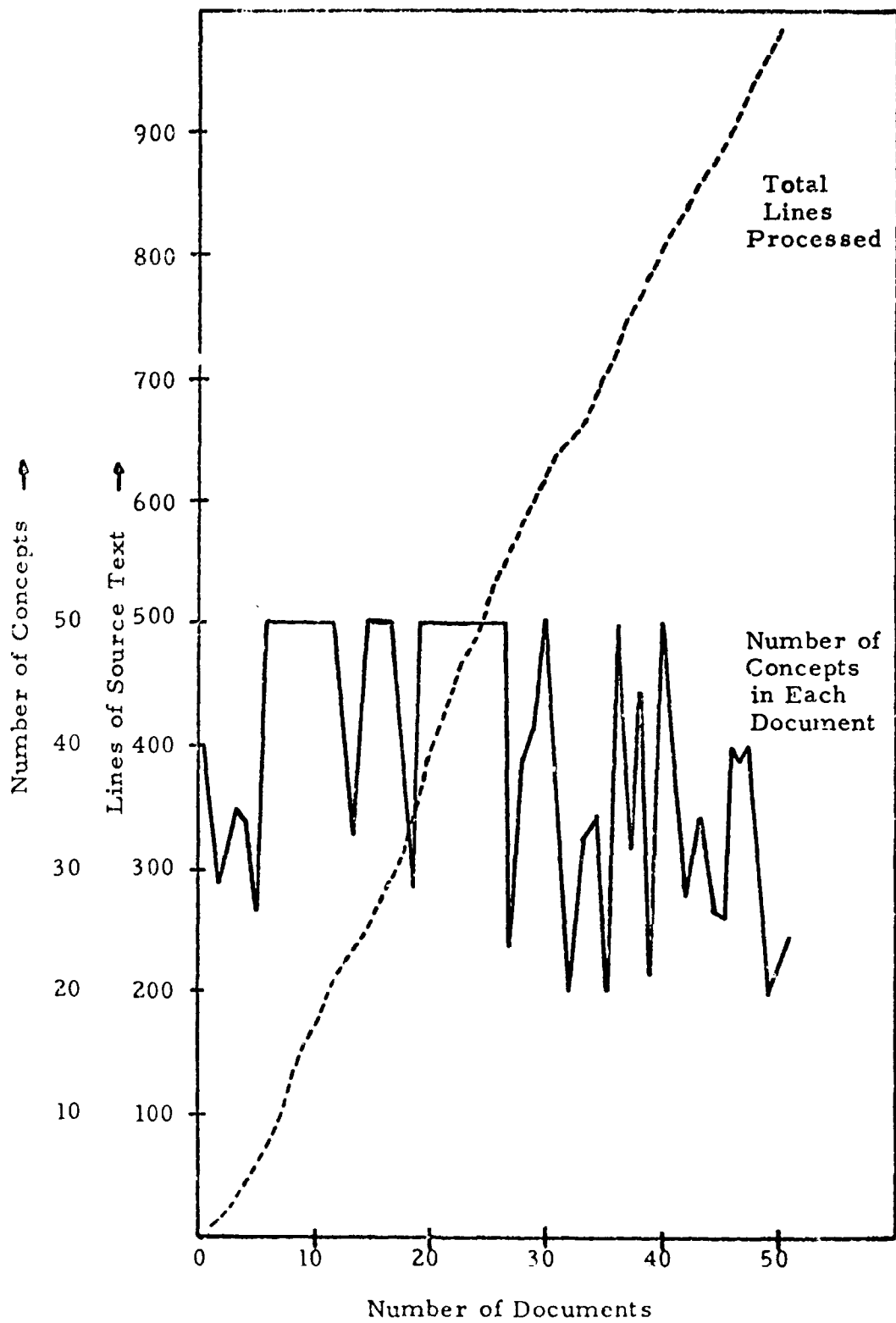


Figure III-8  
Characteristics of Test Document Collection

1	54	0.25000	69	0.25000	152	0.25000	200	0.25000	208	0.25000	208	0.25000
1	209	0.25000	237	0.25000	277	0.25000	337	0.25000	372	0.25000	372	0.25000
1	396	0.25000	403	0.25000	416	0.25000	424	0.25000	437 <sup>a</sup>	0.25000	437 <sup>a</sup>	0.25000
1	483	0.25000	500	0.25000	505	0.25000	507 <sup>a</sup>	0.25000	539	0.25000	539	0.25000
1	555	0.25000	571	0.25000	585	0.25000	587	0.25000	633	0.25000	633	0.25000
1	634	0.25000	651	0.25000	658	0.25000	675	0.25000	711	0.25000	711	0.25000
1	715	0.25000	777	0.25000	781	0.25000	782	0.25000	800	0.25000	800	0.25000
1	811	0.25000	833	0.25000	844	0.25000	855	0.25000	872	0.25000	872	0.25000
2	54	0.75000	131	0.25000	160	0.25000	175	0.25000	197	0.25000	197	0.25000
2	250	0.25000	327	0.50000	332	0.50000	362	0.25000	366	0.25000	366	0.25000
2	403	0.25000	421	0.25000	424	0.25000	539	1.00000	542	0.25000	542	0.25000
2	675	0.25000	711	0.25000	715	0.25000	739	0.75000	742	0.25000	742	0.25000
2	746	0.25000	777	0.25000	781	0.25000	823	0.25000	826	0.25000	826	0.25000
2	838	0.25000	847	0.25000	899	0.25000	0	0.	0	0.	0	0.
3	40	0.33333	46	0.33333	75	0.33333	157	0.33333	175	0.33333	175	0.33333
3	203	0.33333	253	0.33333	258	0.33333	261	0.33333	267 <sup>a</sup>	0.33333	267 <sup>a</sup>	0.33333
3	292	0.33333	309	0.33333	340	0.33333	352	0.33333	359	0.33333	359	0.33333
3	400	0.33333	421	0.33333	426	0.66667	462	0.33333	550	0.33333	550	1.00000
3	670	0.33333	677	0.33333	680	0.33333	722	0.33333	723	0.33333	723	0.66667
3	724	0.33333	766	0.33333	772	0.33333	774	0.33333	778	0.33333	778	0.33333
3	784	0.33333	808	0.33333	852	0.33333	863	0.33333	865	0.33333	865	0.33333
3	900	0.33333	0	0.	0	0.	0	0.	0	0.	0	0.
4	35	0.33333	43	0.16667	58	0.16667	68	0.16667	75	0.16667	75	0.16667
4	112	1.00000	172	0.16667	247	0.16667	266	0.16667	278	0.33333	278	0.16667
4	279	0.16667	311	0.16667	333	0.16667	358	0.16667	414	0.16667	414	0.16667
4	421	0.33333	423	0.16667	443	0.16667	481	0.16667	527	0.16667	527	0.16667
4	557	0.16667	594	0.16667	638	0.16667	639	0.16667	640	0.16667	640	0.16667
4	649	0.16667	650	0.16667	671	0.16667	677	0.16667	724	0.16667	724	0.16667
4	742	0.16667	761	0.16667	813	0.16667	850	0.16667	855	0.16667	855	0.16667
5	43	0.50000	75	0.50000	140	0.50000	220	0.50000	226	0.50000	226	0.50000
5	266	1.00000	292	0.50000	371	0.50000	393	0.50000	407 <sup>a</sup>	0.50000	407 <sup>a</sup>	0.50000
5	421	1.00000	446	1.00000	545	1.00000	560	1.00000	581	0.50000	581	0.50000
5	694	0.50000	697	0.50000	703	1.00000	724	1.00000	729	0.50000	729	0.50000
5	757	0.50000	794	0.50000	862	0.50000	891	0.50000	896	0.50000	896	0.50000

Figure III-9 (cont'd)  
CONCEPTS File

6	17	0.16667	43	0.16667	49	0.33333	75	0.16667	88	0.16667
6	107	0.16667	108	0.16667	156	0.16667	175	0.16667	217	0.16667
6	227	0.16667	230	0.16667	232	0.50000	238	0.33333	266	0.33333
6	284	0.16667	288	0.16667	290	0.16667	293	0.16667	301	0.16667
6	317	0.16667	329	0.16667	345	0.16667	350	0.16667	374	0.33333
6	384	0.16667	401	0.16667	402	0.16667	407	0.16667	419	0.16667
6	421	1.00000	427	0.16667	438	0.16667	513	0.16667	521	0.16667
6	528	0.16667	532	0.16667	557	0.16667	569	0.16667	626	0.16667
6	638	0.33333	647	0.33333	648	0.33333	649	0.16667	655	0.16667
6	656	0.16667	680	0.16667	691	0.16667	699	0.33333	703	0.16667
7	26	0.14286	38	0.14286	55	0.14286	61	0.28571	64	0.14286
7	88	0.14286	101	0.14286	130	0.14286	162	0.14286	180	0.14286
7	195	0.71429	199	1.00000	203	0.14286	204	0.28571	216	0.14286
7	232	0.28571	236	0.14286	294	0.14286	362	0.14286	385	0.14286
7	389	0.14286	407	0.28571	410	0.14286	439	0.14286	457	0.14286
7	460	0.14286	461	0.14286	462	0.14286	469	0.14286	482	0.14286
7	483	0.14286	491	0.14286	505	0.14286	516	0.14286	520	0.14286
7	524	0.14286	557	0.14286	575	0.14286	599	0.14286	700	0.14286
7	723	0.14286	726	0.14286	738	0.14286	773	0.14286	775	0.14286
7	776	0.14286	788	0.14286	797	0.14286	842	0.14286	848	0.14286
8	17	0.33333	18	0.33333	22	0.33333	24	1.00000	61	0.33333
8	75	0.33333	85	0.33333	120	0.33333	124	0.33333	125	0.33333
8	169	0.33333	184	0.33333	187	0.33333	196	0.33333	222	0.33333
8	237	0.66667	240	0.33333	249	0.33333	252	0.33333	266	1.00000
8	282	0.33333	319	0.33333	334	0.33333	407	0.33333	415	0.33333
8	421	0.33333	427	0.33333	433	0.33333	451	0.33333	466	0.33333
8	498	0.33333	507	0.33333	508	0.33333	510	0.33333	520	0.33333
8	548	0.33333	564	0.33333	566	0.33333	574	0.33333	597	0.33333
8	602	0.33333	646	0.33333	647	0.33333	654	0.33333	656	0.66667
		0.33333	688	0.33333	703	0.33333	714	0.33333	716	0.66667
		0.20000	42	0.20000	56	0.20000	92	0.20000	68	0.20000
		0.20000	92	0.20000	92	0.20000	92	0.20000	109	0.20000
		0.20000	92	0.20000	92	0.20000	92	0.20000	199	0.20000
		0.20000	92	0.20000	92	0.20000	92	0.20000	333	0.20000
		0.20000	92	0.20000	92	0.20000	92	0.20000	333	0.20000

Figure III-9 (concluded)  
CONCEPTS File

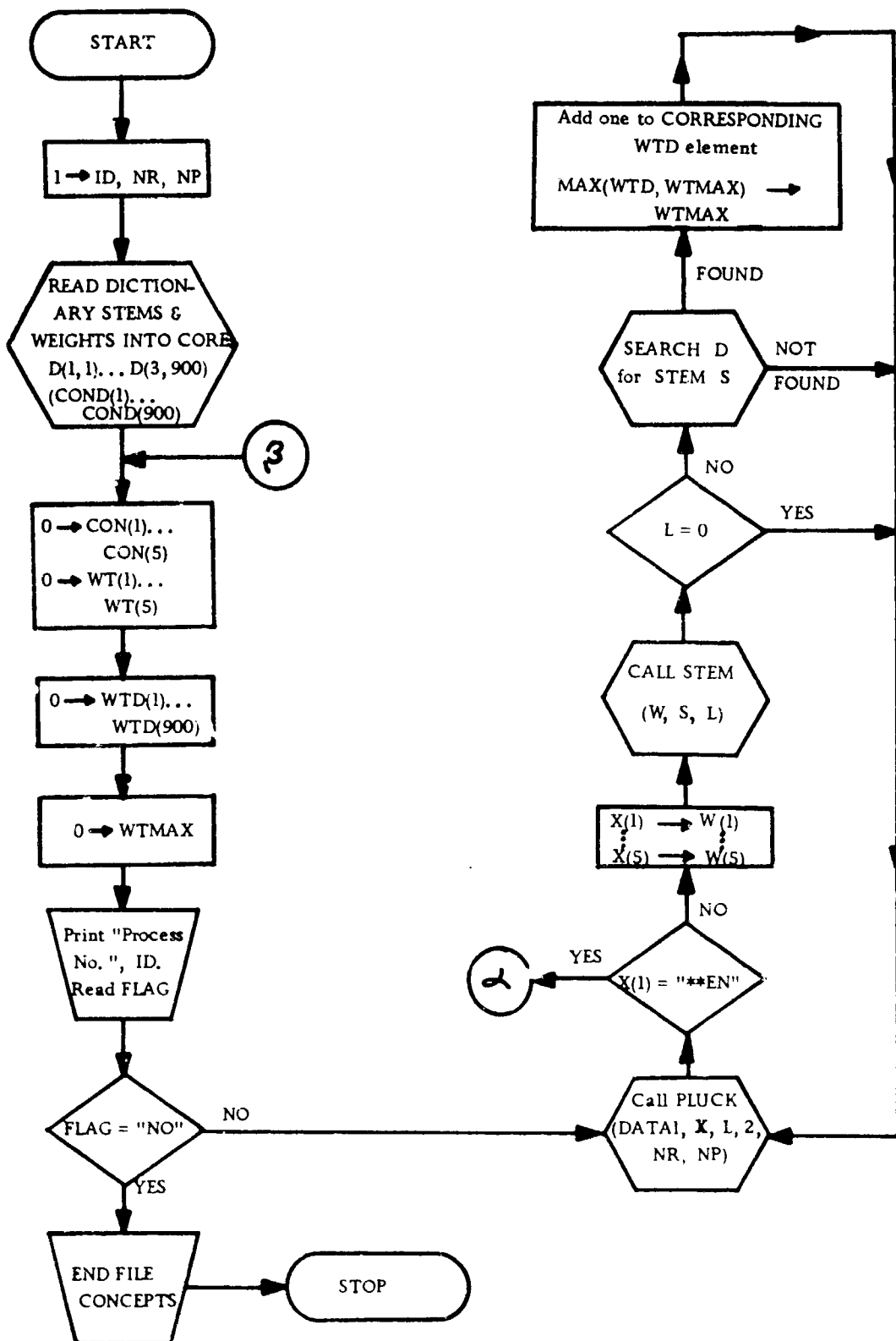


Figure III-10 (cont'd)  
Program CONGRA

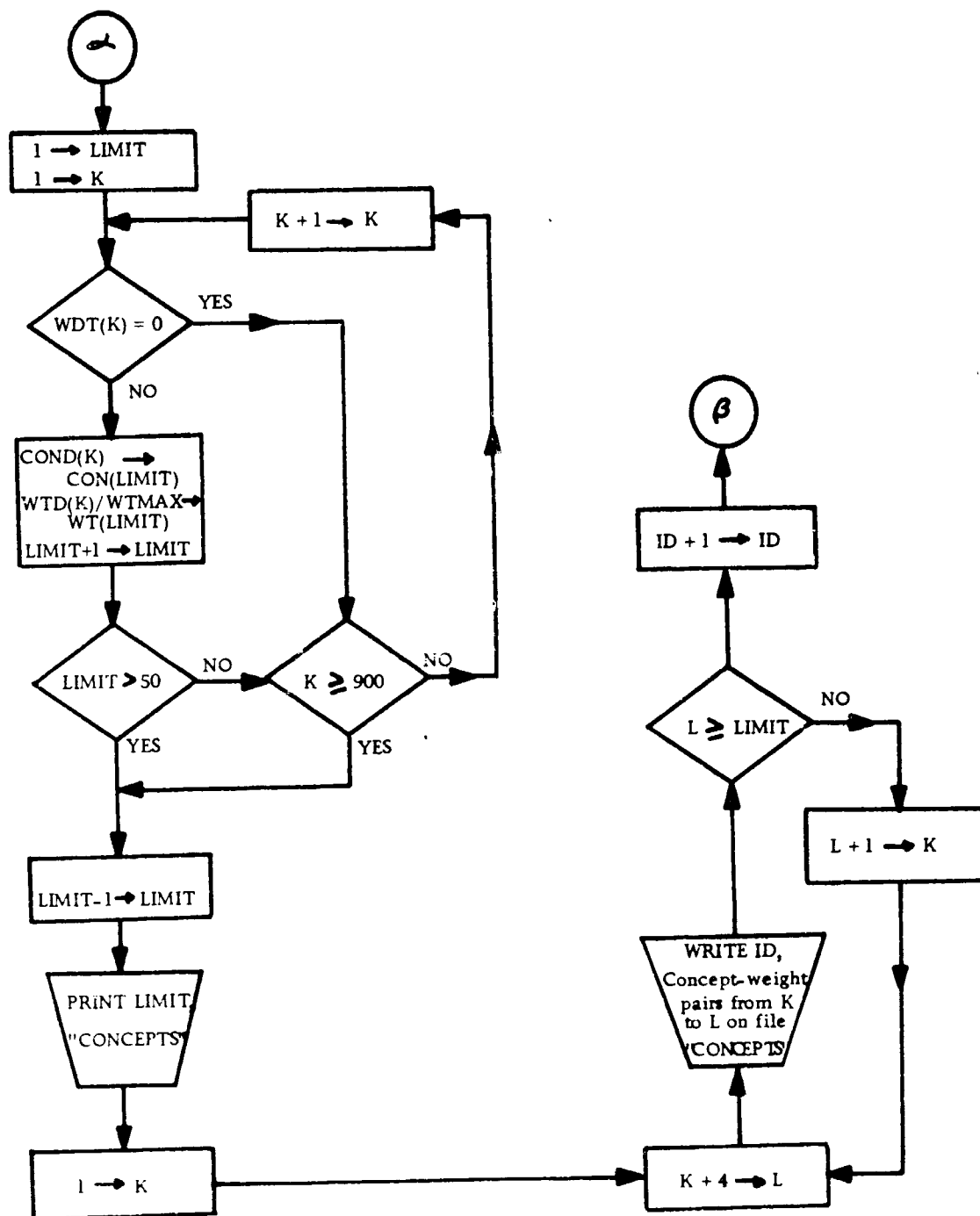


Figure III-10 (concluded)  
Program CONGRA

Let the five digits of a line number be represented as NNTLL.

Then:

NN	=	message number
T	=	0 if the line is a component of the terse form
T	=	1 if the line is a component of the verbose form
LL	=	the number of the line within a multiple-line message.

In order to distinguish between blanks that fill out a line and blanks that are important for spacing, each line of a message is terminated by a vertical arrow.

Figure III-11 is a listing of file MESSAGES.

### III. 2. 6     File OFFLINE

File OFFLINE is used to store documents temporarily for later printing offline on a high-speed printer. If offline printing is requested, file OFFLINE is created, with the desired contents, but it is not printed automatically. The user may "PERM" the file, however.

### III. 2. 7     Core-Resident Files

This subsection first defines the use of certain major variables, arrays and families of arrays, and then shows how several of them are allocated to common storage. Figure III-12(a) lists the families of arrays; Figure III-12(b) lists all the main program variables, and Figure III-12(c) the common variables and arrays.

```

01001 RETRY?;
01101 ANSWER "YES" OR "NO";;
02001 IDENTIFY DOCUMENTS;;
02101 ENTER TEMP ID. (SINGLE OR RANGE), ACC. NO. OR "ALL";;
03101 MORE?;
04001 INVALID.;
04101 REFERENCED DOCUMENTS DO NOT EXIST.;
05001 FORMAT ERROR.;
05101 INCORRECT FORMAT. USE FOR EXAMPLE "13" FOR TEMP. ID. NO.;
05102 13; "13-33" FOR TEMP. ID. NOS. 13 THROUGH 33 INCLUSIVE;
05103 "A13" FOR ACCESSION NO. 13; "ALL" FOR ALL DOCUMENTS;
05104 RETRIEVED DURING THIS QUERY SEQUENCE.;
06001 TEMP. FILE EMPTY.;
06101 NO DOCUMENTS IN TEMPORARY FILE. THEREFORE, ONLY ACCESSION;
06102 NUMBERS CAN BE USED TO SPECIFY DOCUMENTS.;
07101 IS NORMAL OPERATION DESIRED?;
08001 SKIP INITIAL?;
08101 SKIP INITIAL QUERY?;
09101 ENTER WORDS FOR INITIAL SEARCH QUERY, FOLLOWED BY "END";;
10001 WORDS NOT IN DICTIONARY;
10101 THE FOLLOWING WORDS ARE NOT USEFUL FOR RETRIEVAL FROM THIS;
10102 COLLECTION;
11001 NO WORDS-RETRIEVAL ABORTED.;
11101 NO USEFUL WORDS REMAIN. ANOTHER INITIAL QUERY IS REQUIRED.;
12001 SAME MODE?;
12101 DO YOU WISH TO CONTINUE IN THE SAME MODE?;
13001 QUIT?;
13101 DO YOU WISH TO TERMINATE USE OF THE SYSTEM?;
14101 MODE FLAGS ALL OFF. IDENTIFY NUMBERS OF FLAGS TO BE SET ON;
14102 FOLLOWED BY "END";;
15001 PRINT PRSM?;
15101 DO YOU WANT AN ANALYSIS OF THE WORDS (IF ANY) IN THE QUERY?;
16101 ;
16102 WORD
16103 ;

```

STEP: CONCEPT-WEIGHT PAIRS

Figure III-11 (cont'd)  
MESSAGES File

```

17001 PRINT QUERY VECTOR?:
17101 DO YOU WANT TO SEE THE QUERY CONCEPT VECTOR?:
18101 ,
18102 CON- WEIGHT CON- WEIGHT CON- WEIGHT CON- WEIGHT
18103 CEPT CEPT CEPT CEPT
18104 ,
19001 NO. OF HITS >= 50.
19101 AT LEAST 50 DOCUMENTS MEET THE SPECIFICATIONS FOR THE PRESENT
19102 QUERY.
20001 NO. OF HITS=
20101 THE NUMBER OF DOCUMENTS MEETING YOUR SPECIFICATIONS FOR THE
20102 QUERY IS
21001 PRINT TEMP?:
21101 DO YOU WANT A PRINTOUT OF THE TEMPORARY FILE?:
22101 ,
22102 ACCESSION TEMPORARY CORRELATION RANK BIB. DATA PRINTED/
22103 NUMBER IDENT. [WHEN LAST RETRIEVED] RET'VD LAST QUERY
22104 ,
23001 PRINT BIBLIO.?:
23101 DO YOU WANT BIBLIOGRAPHIC INFORMATION FOR SOME OF THE RETRIEVED
23102 DOCUMENTS?:
24001 **
24101 PREVIOUSLY PRINTED.
25001 THAT'S ALL.
25101 BIBLIOGRAPHIC DATA FOR ALL THE TEMPORARY FILE DOCUMENTS HAVE
25102 BEEN PRINTED.
26101 ON-LINE RETRIEVAL SYSTEM SIGNING OFF.
26102 , -
27001 OPTIONS:
27101 ENTER OPTION NAME (OR "HELP" TO SEE THE LIST OF AVAILABLE
27102 OPTIONS):
28001 OPTIONS: "END", "MOD", "DOC".
28101 OPTIONS AVAILABLE ARE:
28102 ,
28103 "END" - TERMINATE THIS SEARCH QUERY SEQUENCE FOR STARTING

```

Figure III-11 (cont'd)  
MESSAGES File

```

28104      A NEW SEQUENCE OR SIGNING OFF.†
28105      †
28106      "MOD" - MODIFY OR REPLACE THE PRESENT QUERY AND CONTINUE†
28107      THE PRESENT QUERY SEQUENCE.†
28108      †
28109      "DOC" - PRINT DATA FOR DOCUMENTS RETRIEVED DURING THIS†
28110      SEQUENCE OR ANY DOCUMENTS OF KNOWN ACCESSION NUMBER.†
28111      †
29001      "OFF", "CHG", "CON", "RET", "DEL", "SEE", "CLR", "WRD", "DDC", "WGT".†
29101      "OFF" - CREATE FILE FOR OFFLINE DOCUMENT PRINTING†
29102      "CHG" - CHANGE MODE OF OPERATION (QUERY SEQUENCE TERMINATION†
29103      NOT REQUIRED.) A LIST OF MODES IS PROVIDED.†
29105      "CON" - INSPECT THE CONCEPT VECTORS OF DOCUMENTS.†
29107      *"RET" - EXECUTE THE PRESENT RETRIEVAL REQUEST.†
29109      *"DEL" - DELETE UNWANTED DOCUMENTS RETRIEVED DURING THE†
29110      PRESENT QUERY SEQUENCE.†
29112      *"SEE" - INSPECT THE EXISTING QUERY.†
29114      *"CLR" - ERASE THE EXISTING QUERY.†
29116      *"WRD" - ADD OR DELETE QUERY WORDS.†
29119      *"DDC" - PERFORM DOCUMENT-DOCUMENT CORRELATION.†
29121      *"WGT" - PERFORM DIRECT MANIPULATION OF QUERY CONCEPT VECTORS.†
29122      †
29123      (OPTIONS MARKED WITH "*" ARE NORMALLY CALLED AUTOMATICALLY FOR†
29124      THE USER BY THE SYSTEM.)†
29125      †
30001      MORE?†
30101      OTHER OPTIONS ARE AVAILABLE. DO YOU WANT A LIST?†
31001      INVALID.†
31101      THE OPTION NAME YOU ENTERED DOES NOT EXIST.†
32001      SEQUENCE KILLED.†
32101      PRESENT SEARCH QUERY SEQUENCE TERMINATED. A NEW QUERY MAY†
32102      BE INITIATED AT THIS TIME OR YOU MAY SIGN OFF.†
33001      CONTINUE?†
33101      CONTINUE PRINTING FROM THIS SPECIFIED GROUP?†
34101      DO YOU WANT AN EXPLANATION OF THE AVAILABLE MODES?†

```

Figure III-11 (cont'd)  
MESSAGES File

35001 1=TERSE; 2=SKIP INITIAL; 3=QUERY ANALYSIS; 4=RETRIEVAL ANALYSIS;†  
 35002 5=ALL OPTIONS.†  
 35101 MODES ARE NORMALLY "OFF" AND CAN BE TURNED ON BY TYPING IN A FLAG.†  
 35102 NUMBER OR SEQUENCE OF NUMBERS, SUCH AS "1,3,5,END". THE FOLLOWING  
 35103 MODES ARE AVAILABLE:†  
 35104 FLAG NUMBER ACTION†  
 35106 ,  
 35107 1 SELECT TERSE DIALOGUE.†  
 35109 2 SKIP FORMATION OF INITIAL QUERY IN QUERY†  
 35110 SEQUENCE FROM WORDS.†  
 35111 3 MAKE AVAILABLE QUERY WORDS, STEMS AND CONCEPTS†  
 35112 BEFORE RETRIEVAL.†  
 35113 4 MAKE AVAILABLE TEMP TABLE CONTENTS AFTER†  
 35114 RETRIEVAL.†  
 35115 5 ASSUME ANY OPTION MAY BE USED.†  
 36001 DELETE ACTIVE.†  
 36101 SOME DOCUMENTS ARE TO BE DELETED FROM THE TEMPORARY FILE--†  
 37001 BEFORE RETRIEVING.†  
 37101 BEFORE THE PRESENT RETRIEVAL IS PERFORMED.†  
 38001 SELECT.†  
 38101 YOU MUST SELECT THE DOCUMENTS TO BE DELETED.†  
 39001 ,  
 39101 ACCESSION NO. IS†  
 40100 [FORM SPACE OVER -- ADD LATER]†  
 40101 DATA SAVED IN FILE "OFFLINE".†  
 40102 ,  
 41001 TEMP. FILE EMPTY TO START.†  
 41101 TEMPORARY FILE EMPTY PRIOR TO EXECUTION OF PRESENT QUERY.†  
 42001 TEMP ID. STARTS WITH†  
 42101 DOCUMENTS FOUND BY THIS RETRIEVAL WILL HAVE TEMP. NOS.†  
 42102 STARTING WITH ,  
 43001 EXCLUDE PREVIOUS?;†  
 43101 SHOULD DOCUMENTS RETRIEVED PREVIOUSLY DURING THIS QUERY SEQUENCE†  
 43102 BE EXCLUDED FROM RE-RETRIEVAL?;†  
 44001 SUPPRESS PREV. PRINTED?;†

Figure III-11 (cont'd)  
MESSAGES File

44101 SHOULD PRINTING OF BIBLIOGRAPHIC DATA PREVIOUSLY PRINTED BE?  
 44102 SUPPRESSED?:  
 45001 SPACES IN TEMP. WANT MORE?:  
 45101 SPACES EXIST IN THE TEMPORARY FILE FOR NEW RETRIEVALS. IS MORE?  
 45102 SPACE DESIRED?:  
 46001 TABLE FULL.  
 46101 TABLE OF DOCUMENTS RETRIEVED DURING THIS QUERY SEQUENCE IS FULL.  
 46102 SPACE MUST BE MADE BEFORE EXECUTING ANOTHER QUERY.  
 47001 TEMP. DOCS. ONLY, SHORT FORM?:  
 47101 PRINT ONLY RANKING AND BIBLIOGRAPHIC DATA (NO ABSTRACTS) FOR  
 47102 DOCUMENTS RETRIEVED DURING THIS QUERY SEQUENCE, EXCLUDING  
 47103 BIBLIOGRAPHIC DATA ALREADY PRINTED?  
 48001 ON NO-N0 LIST.  
 48101 THIS DOCUMENT EXCLUDED FROM RE-RETRIEVAL DURING PRESENT QUERY  
 48102 SEQUENCE.  
 49001 PRINT BIBLIOGRAPHIC AGAIN?:  
 49101 BIBLIOGRAPHIC DATA PRINTED BEFORE DURING THIS QUERY SEQUENCE.  
 49102 PRINT AGAIN?:  
 50101 PRINT ABSTRACT?:  
 51010 DOC.-DOC.?:  
 51101 DO YOU WANT TO ERASE THE PRESENT QUERY AND DO DOCUMENT-DOCUMENT?  
 51102 SEARCHING?:  
 52001  
 52101 NOW SPECIFY THE DOCUMENTS FOR CORRELATION.  
 53001 QUERY WORD ACTION?:  
 53101 DO YOU WANT TO SEE OR MODIFY THE WORDS FORMING THE QUERY?:  
 54001 PRESENT:  
 54101 THE PRESENT QUERY WORDS ARE:  
 55001 ADD WORDS?:  
 55101 DO YOU WANT TO ADD OR REPLICATE ANY WORDS?:  
 56001 ENTER WORDS:  
 56101 ENTER WORDS, FOLLOWED BY "END":  
 57001 DELETE?:  
 57101 DO YOU WANT TO DELETE ANY WORDS?:  
 58001 NOT IN QUERY.

Figure III-11 (cont'd)  
MESSAGES File

58101 YOU CANNOT DELETE A WORD THAT IS NOT ALREADY IN THE QUERY.:  
59001 CONTINUE PREV.?:  
59101 DO YOU WANT TO BUILD ON THE PREVIOUS DOCUMENT-DOCUMENT SEARCH?:  
60001 DIRECT CON. VECT. ACTION?:  
60101 DO YOU WANT TO INSPECT OR DIRECTLY MODIFY THE QUERY CONCEPT:  
60102 VECTOR?:  
61001 MODIFY?:  
61101 DO YOU WANT TO MODIFY THIS VECTOR?:  
62001 ENTER PAIR:  
62101 ENTER CONCEPT-WEIGHT PAIR (C.G. "1203, -0.123"):  
63101 ENTER PAIR:  
64001 INVALID.  
64101 INVALID CONCEPT NUMBER. DO YOU WANT TO TRY ANY :  
65001 RETRIEVE?:  
65101 DO YOU WANT A RETRIEVAL PERFORMED WITH THE PRESENT QUERY VECTOR?:  
66001 NULL VECTOR. WANT NEW SEQUENCE?:  
66101 A RETRIEVAL CANNOT BE PERFORMED BECAUSE YOUR PRESENT QUERY VECTOR:  
66102 IS NULL. DO YOU WANT TO START A NEW QUERY SEQUENCE?:  
67001 OFF LINE PRINT.:  
67101 READY TO PRINT DOCUMENTS OFF LINE.:  
68001 :  
68101 SPECIFY FIRST DOCUMENT OR DOCUMENT GROUP TO PRINT.:  
69001 MORE DOC.-DOC.?:  
69101 DO YOU WANT TO PERFORM MORE DOCUMENT-DOCUMENT SEARCHING?:  
70001 TOTALLY REPLACE PRESENT QUERY?:  
70101 DO YOU WANT TO ERASE COMPLETELY YOUR PRESENT QUERY AND ENTER:  
70102 NEW QUERY WORDS?:  
71001 PRESENT QUERY?:  
71101 THE PRESENT QUERY CONCEPT VECTOR IS:  
72001 CLEARED.:  
72101 THE PAST QUERIES AND QUERY CONCEPT VECTORS HAVE BEEN CLEARED.:  
73101 ILLEGAL SELECTION; REQUEST IGNORED.:  
74101 :  
74102 :  
74103

Figure III-11 (concluded)  
MESSAGES File

Name	Contents
TEMP	Temporary file: accession number, temporary identification number, correlation coefficient and rank when last retrieved, print suppression flag and flag indicating if retrieved on last executed retrieval.
PRE	Words for queries: the words, their stems and concept-weight pair mappings.
QUERY	The query concept vector.

Figure III-12(a)  
Families of Arrays

Name	Type	Use
IX	I	Next available temporary identification number.
JX	I	Number of entries presently in TEMP.
NEWQ	L	Mode setting precedes initial query.
RFLG	L	Present query not initial.
DEFLG	L	DEL entered through RET.
WFLG	L	MOD has altered PRE.
SEEFLG	L	SEE activated by MOD.
WRDFLG	L	WRD activated by MOD.
DOCDOC	L	Last retrieval in present sequence used document-document correlation.
TERSE *	L	Terse dialogue: Mode 1 selected.
SKIPI	L	Skip initial query: Mode 2 selected.
PRINTQ	L	QUERY available immediately before retrieval: Mode 3 selected.
OPTION	L	HELP prints all options: Mode 5 selected.
LGSTACNO *	I	Largest document accession number in the collection.

\* in common storage

Figure III-12(b)  
Program Variables and Arrays

Name	Dimension	Type	Use
TEMP1	50	I	accession number of document in temporary file
TEMP2	50	I	temporary identification number of document in temporary file
TEMP3	50	F	correlation coefficient of document in temporary file when last retrieved
TEMP4	50	I	rank of document in temporary file when last retrieved
TEMP5	50	L	set when printing of bibliographic data for document in temporary file is to be suppressed
TEMP6	50	L	set when the last executed retrieval command retrieved this document
QUERY1	50	I	concept number for a component of the present query vector
QUERY2	50	F	concept weight for a component of the present query vector
NONO	100	I	accession number of documents for which retrieval is suppressed
TERSE	---	L	set if terse mode is selected
LGSTACNO	---	I	largest accession number in document file
PRE1	5 x 25	A	present query word
PRE2	3 x 25	A	present query stem
PRE3	3 x 25	I	concept numbers for stem
PRE4	3 x 25	F	concept weights for stem

Figure III-12(c)  
Common Storage

### III.3 FLOWCHART OF THE DIALOGUE PROCESSOR

Figure III-13 contains a flowchart of the dialogue processor, whose operation is described in subsection III.1.1.

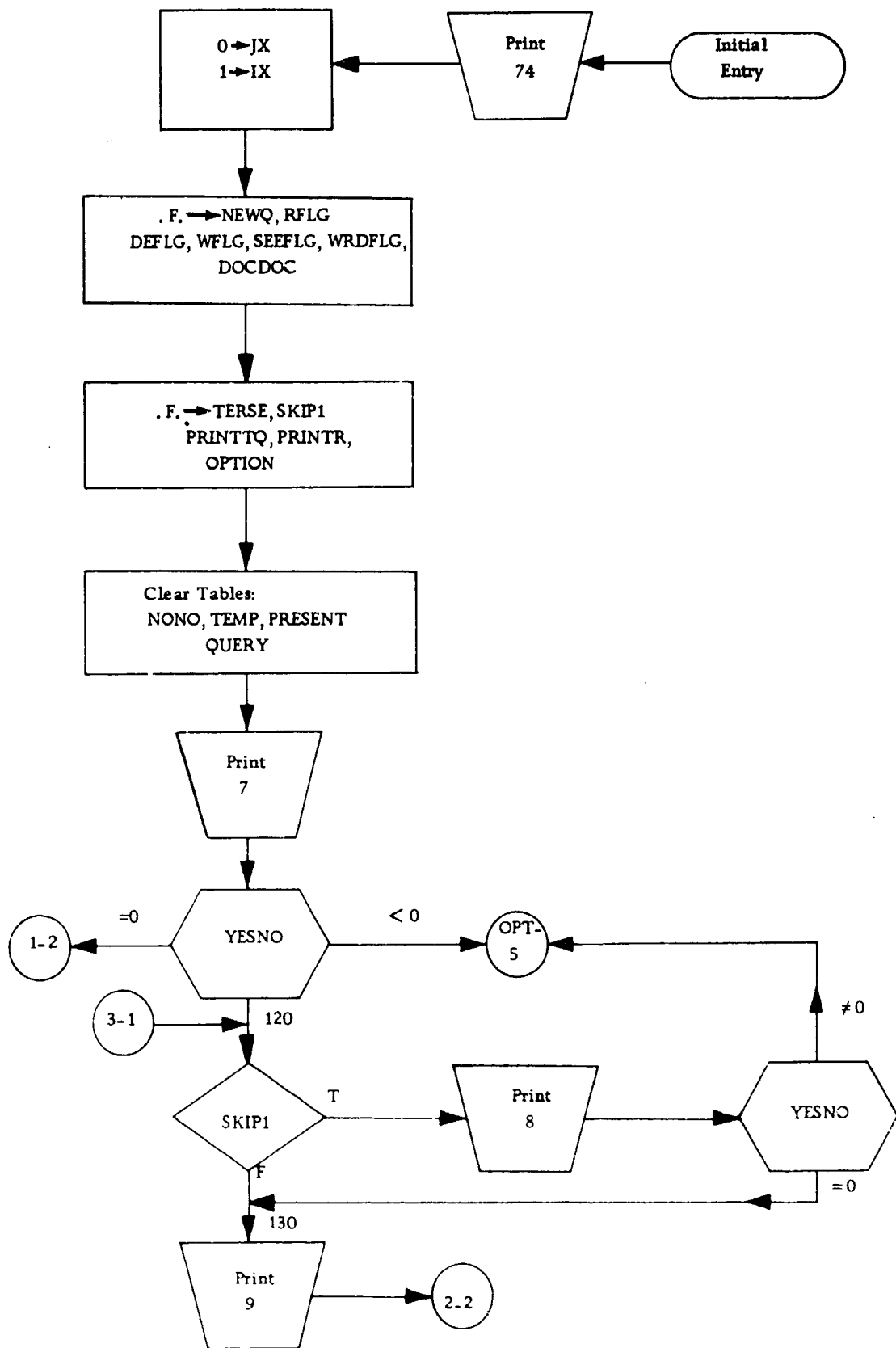


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR

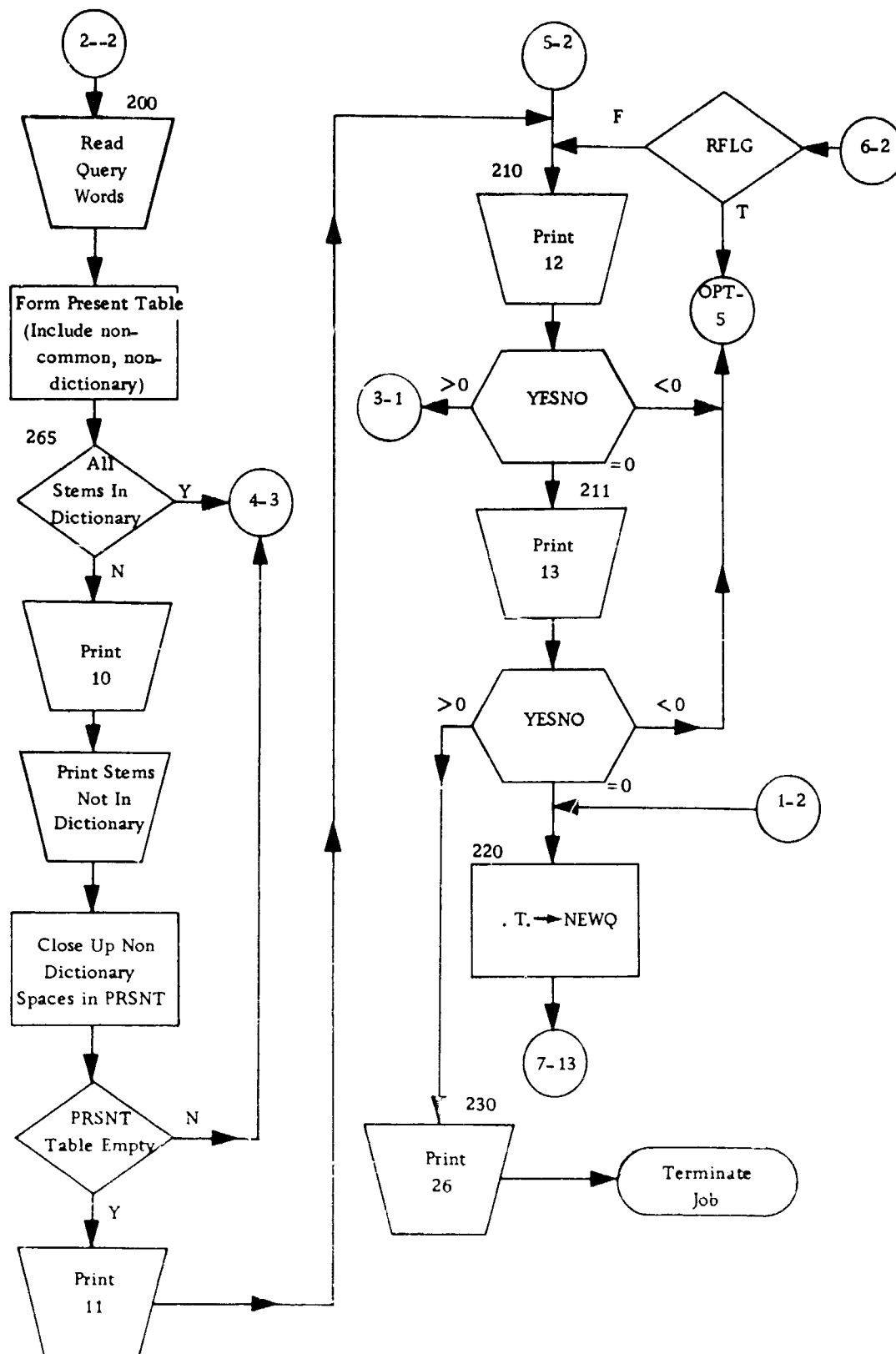


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR

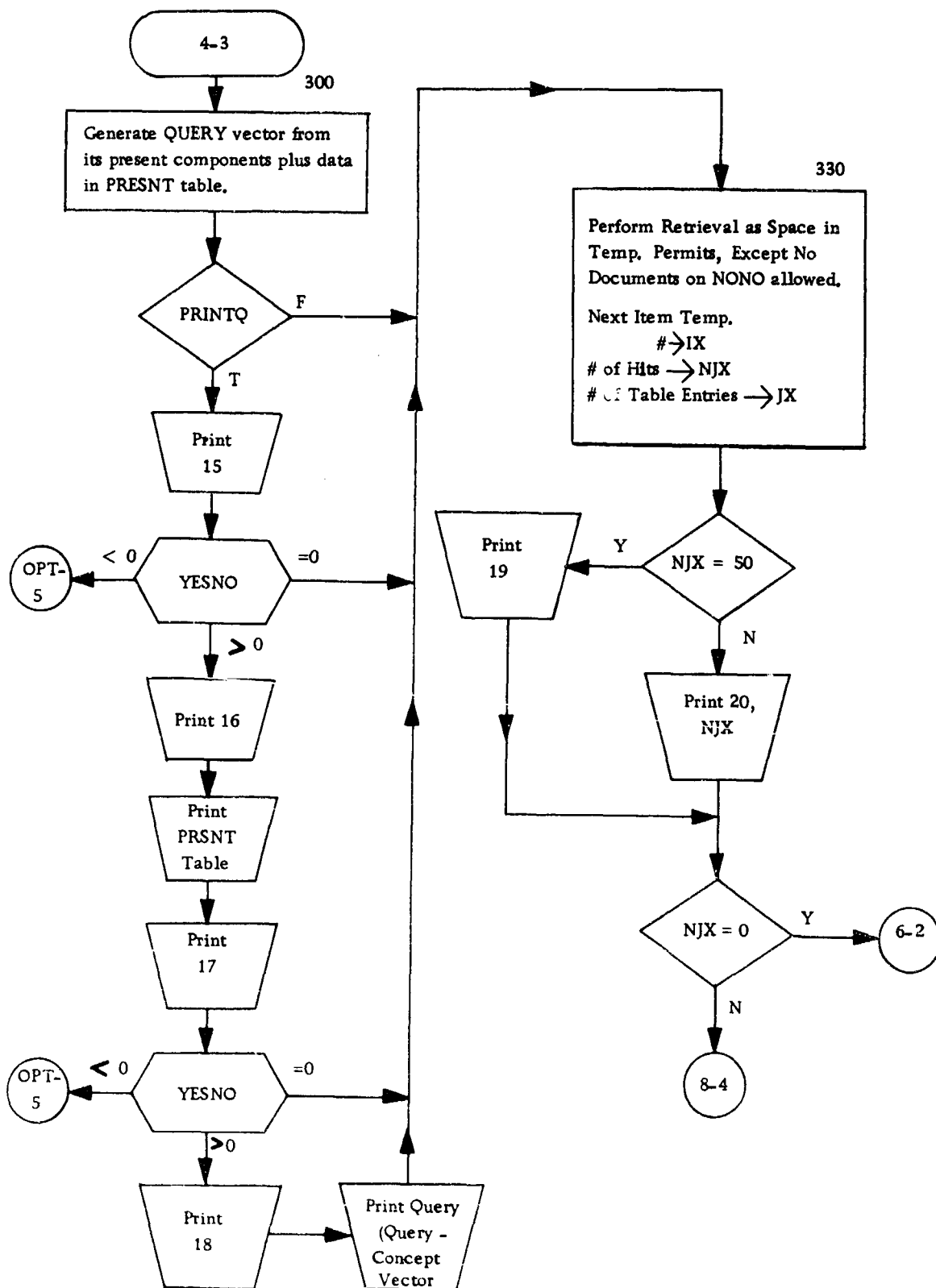


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR

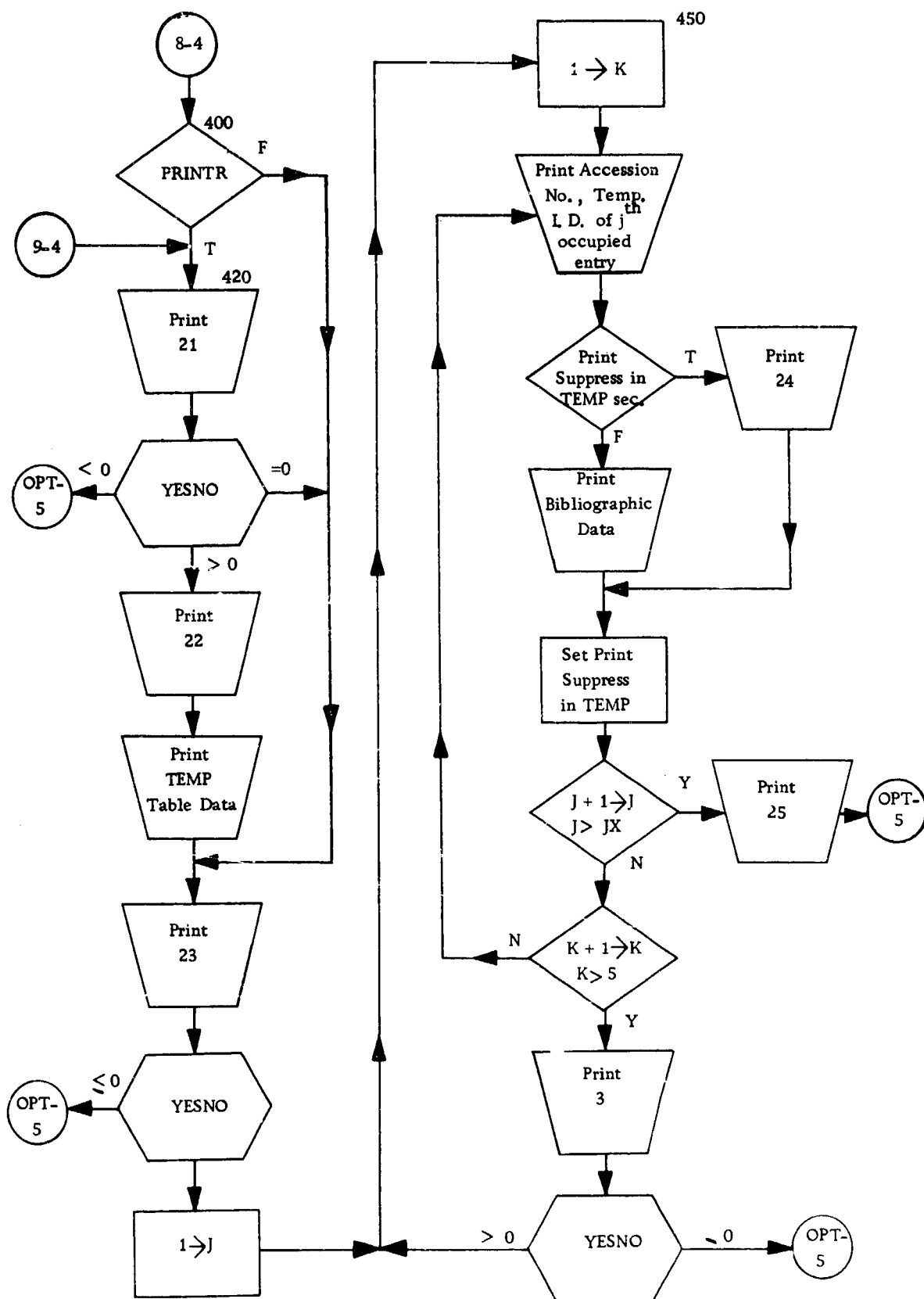


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR

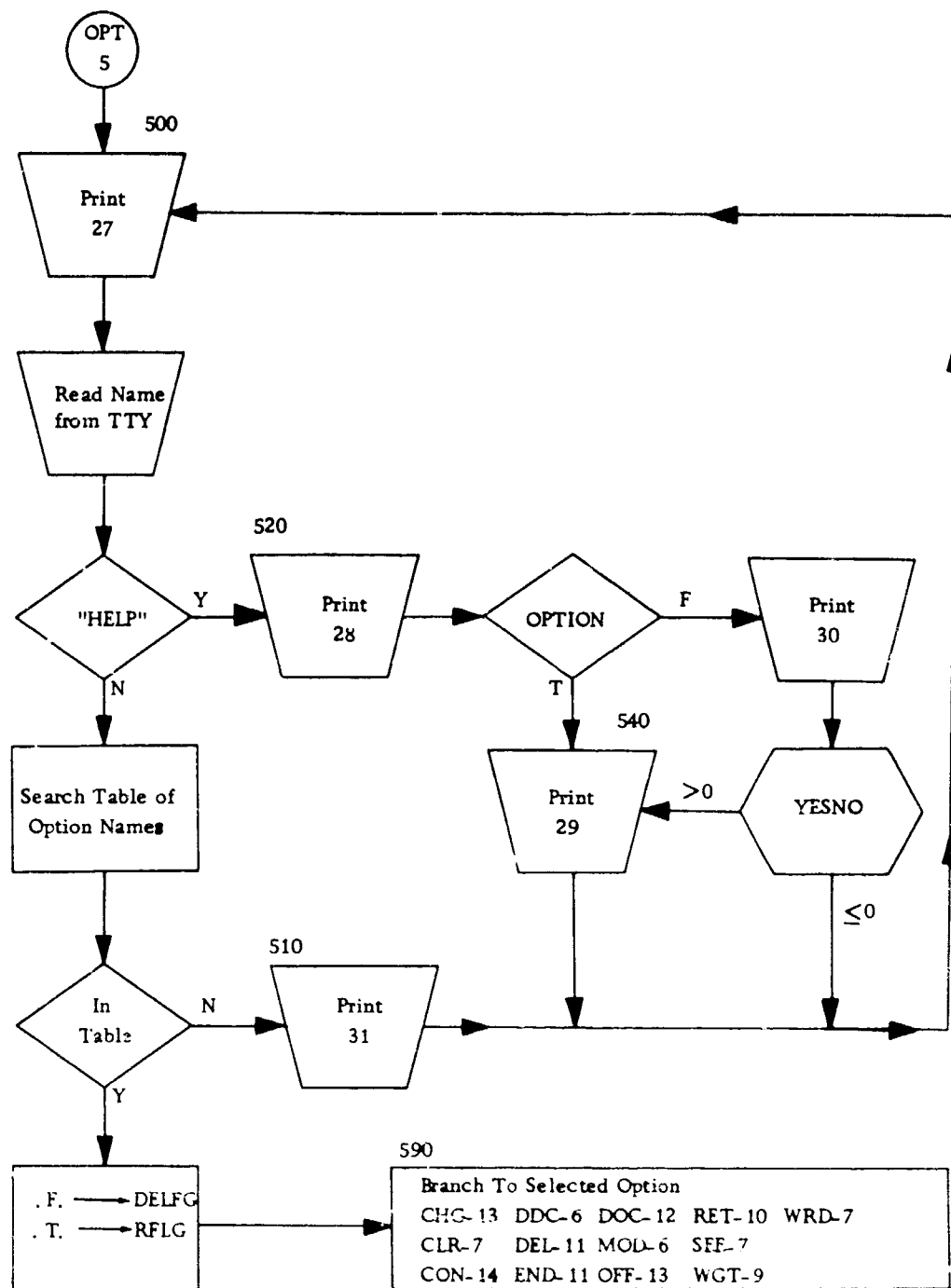


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR

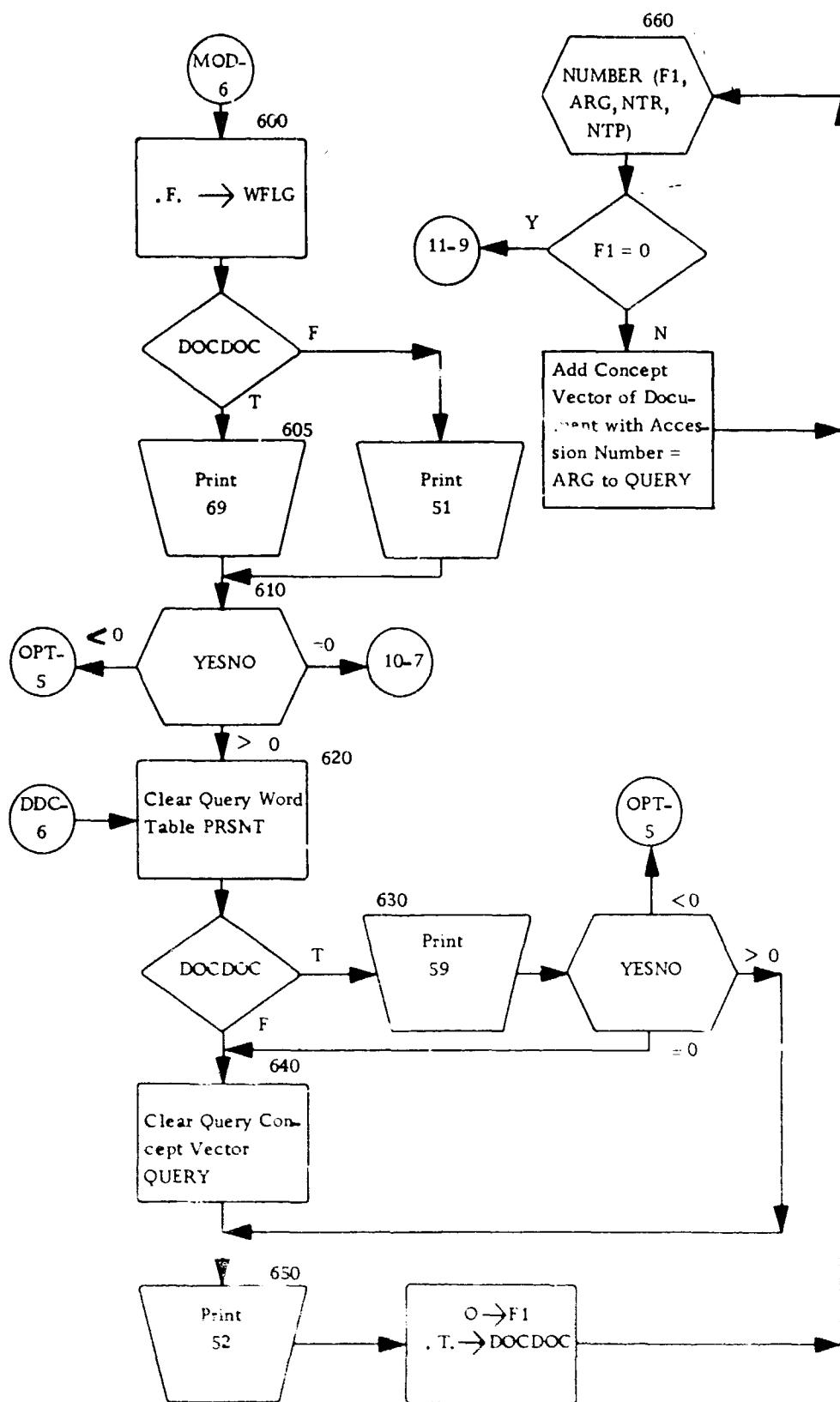


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR

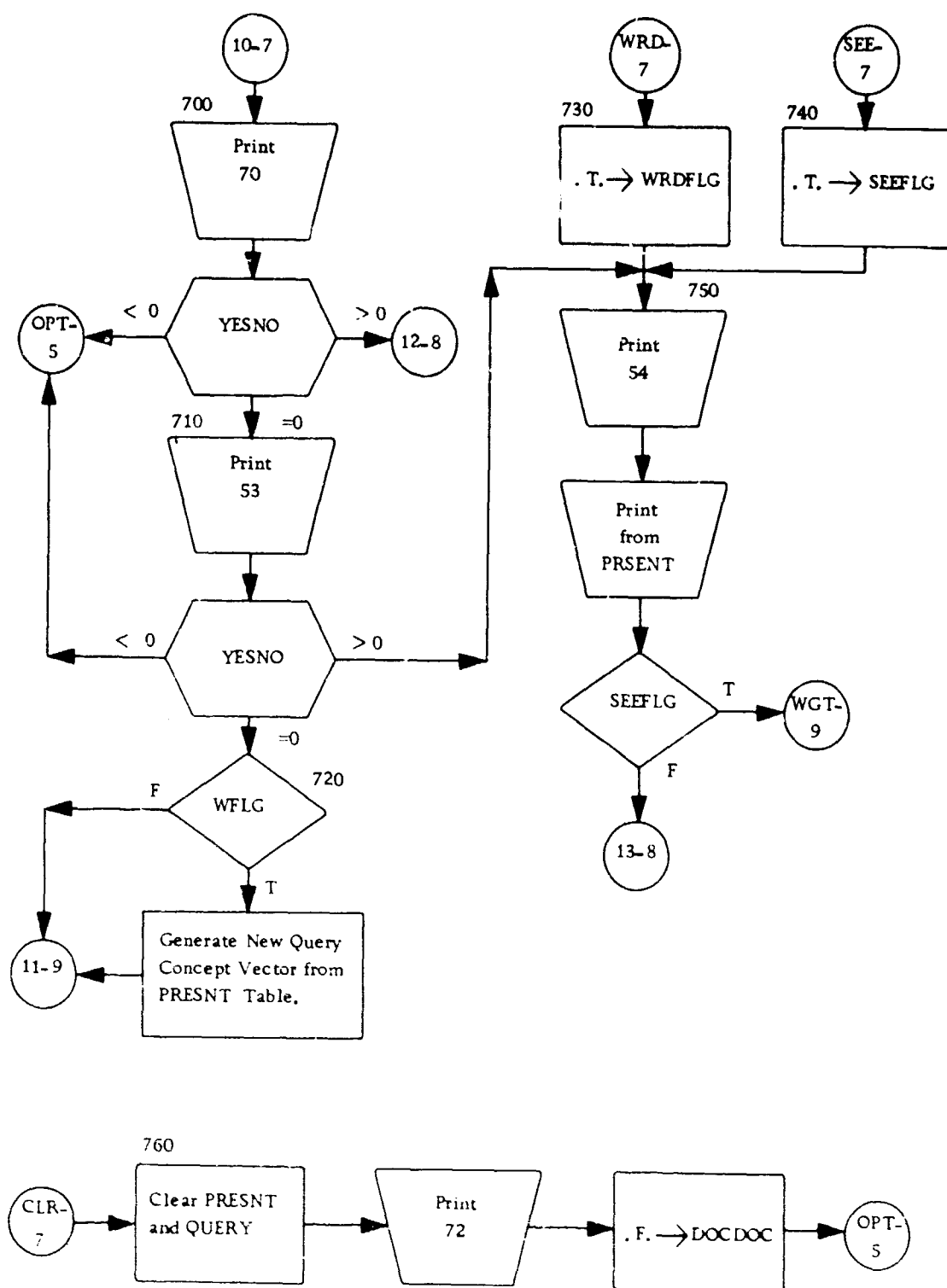


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR

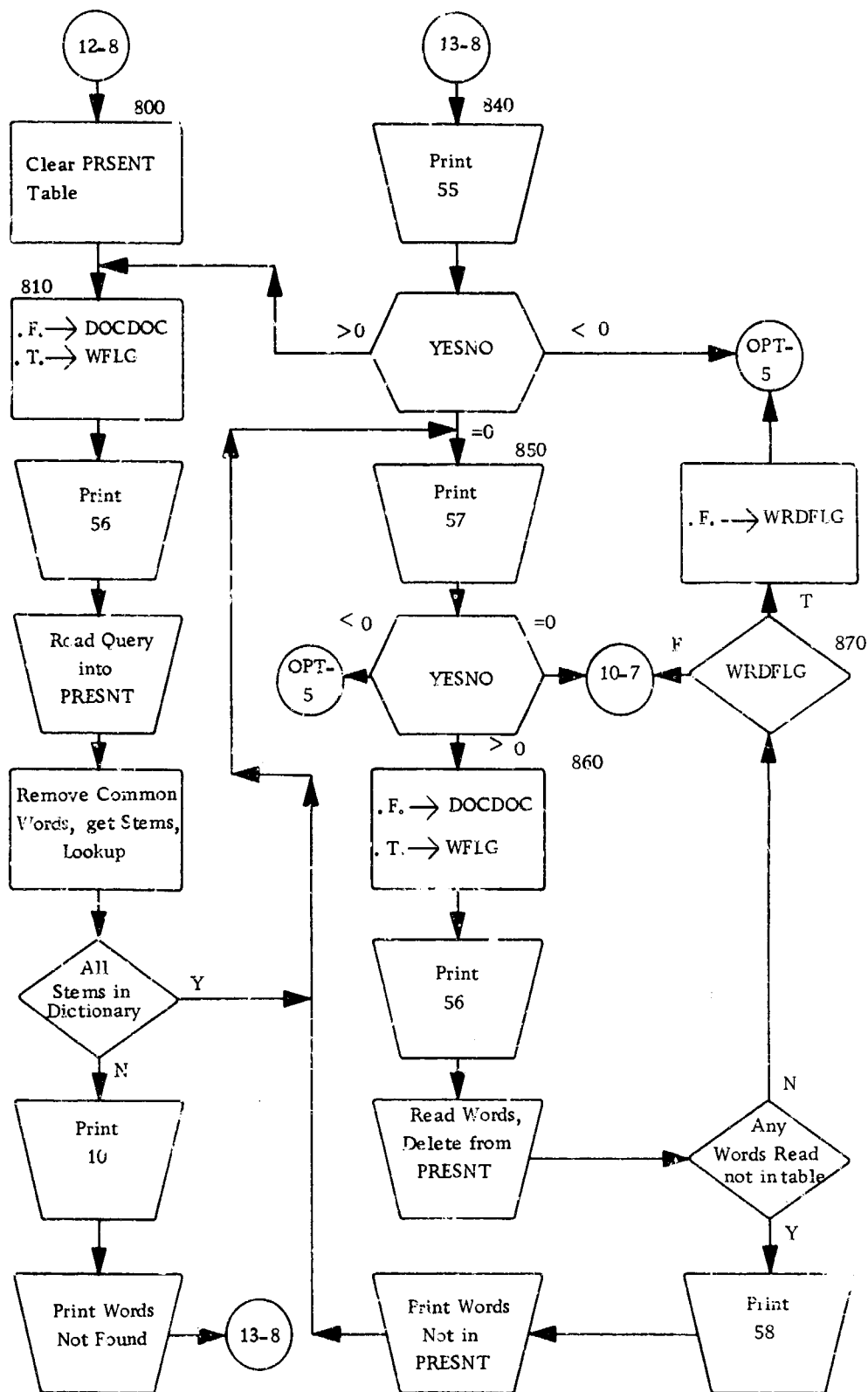


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR  
III-88

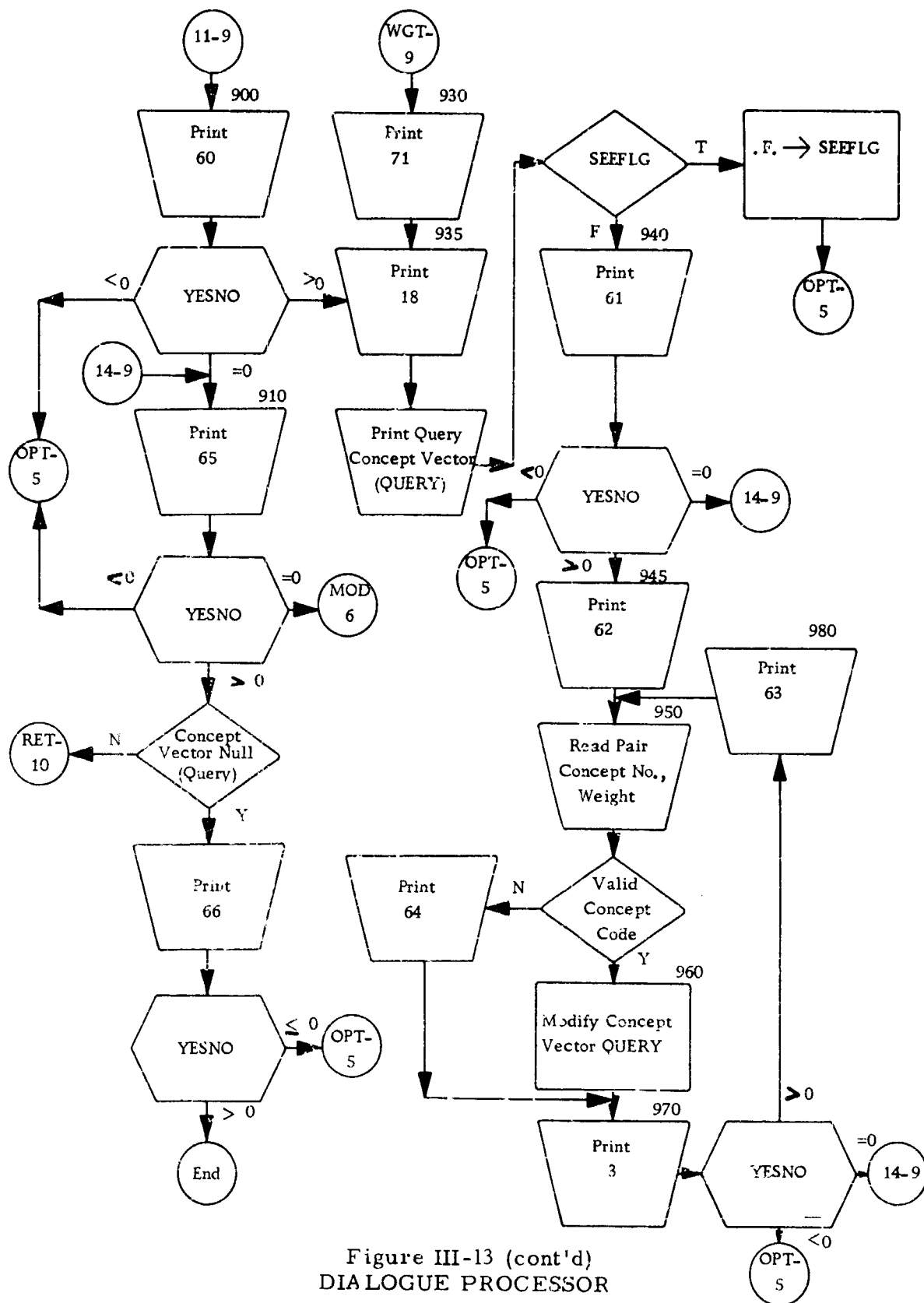


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR

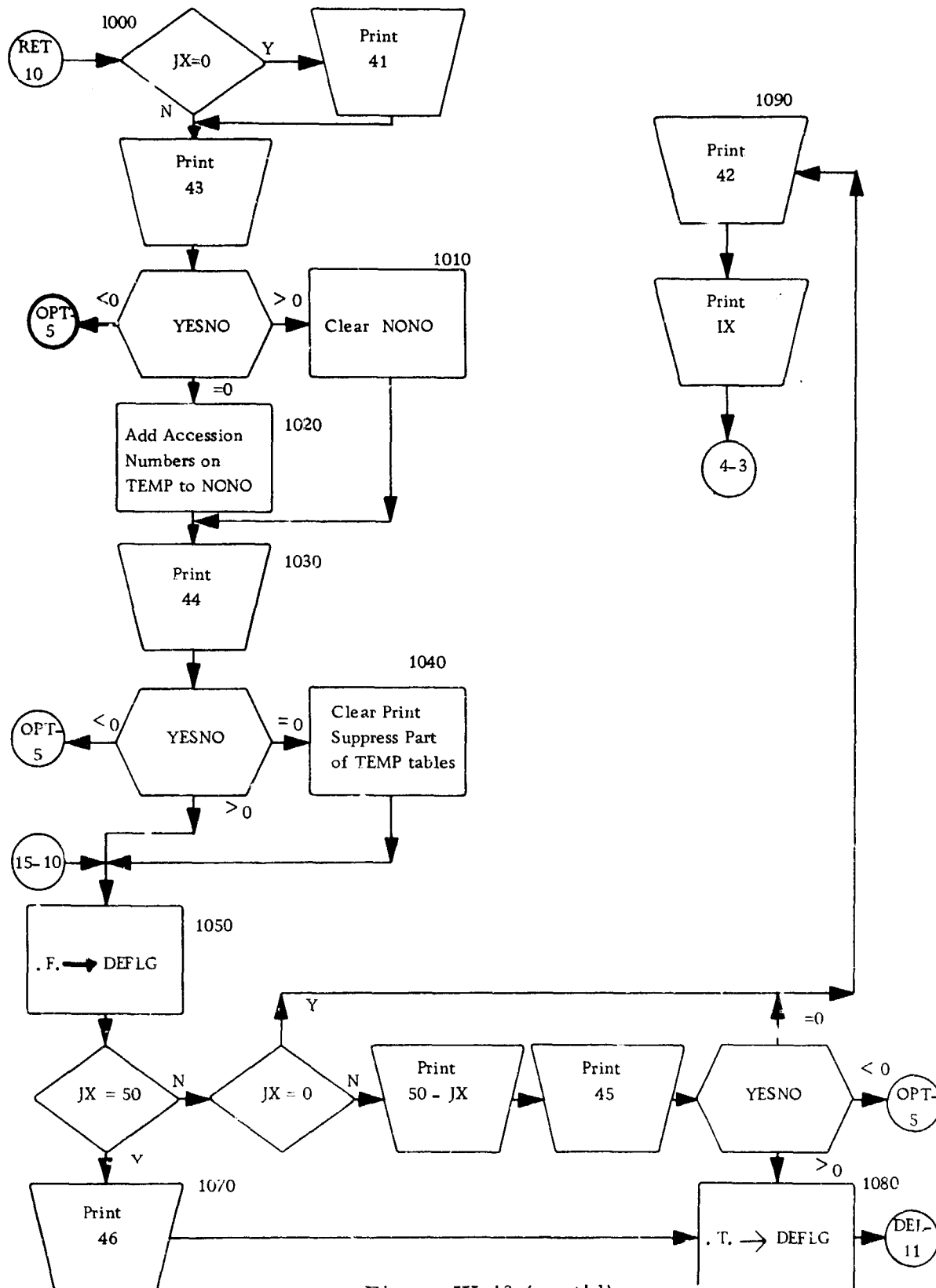


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR

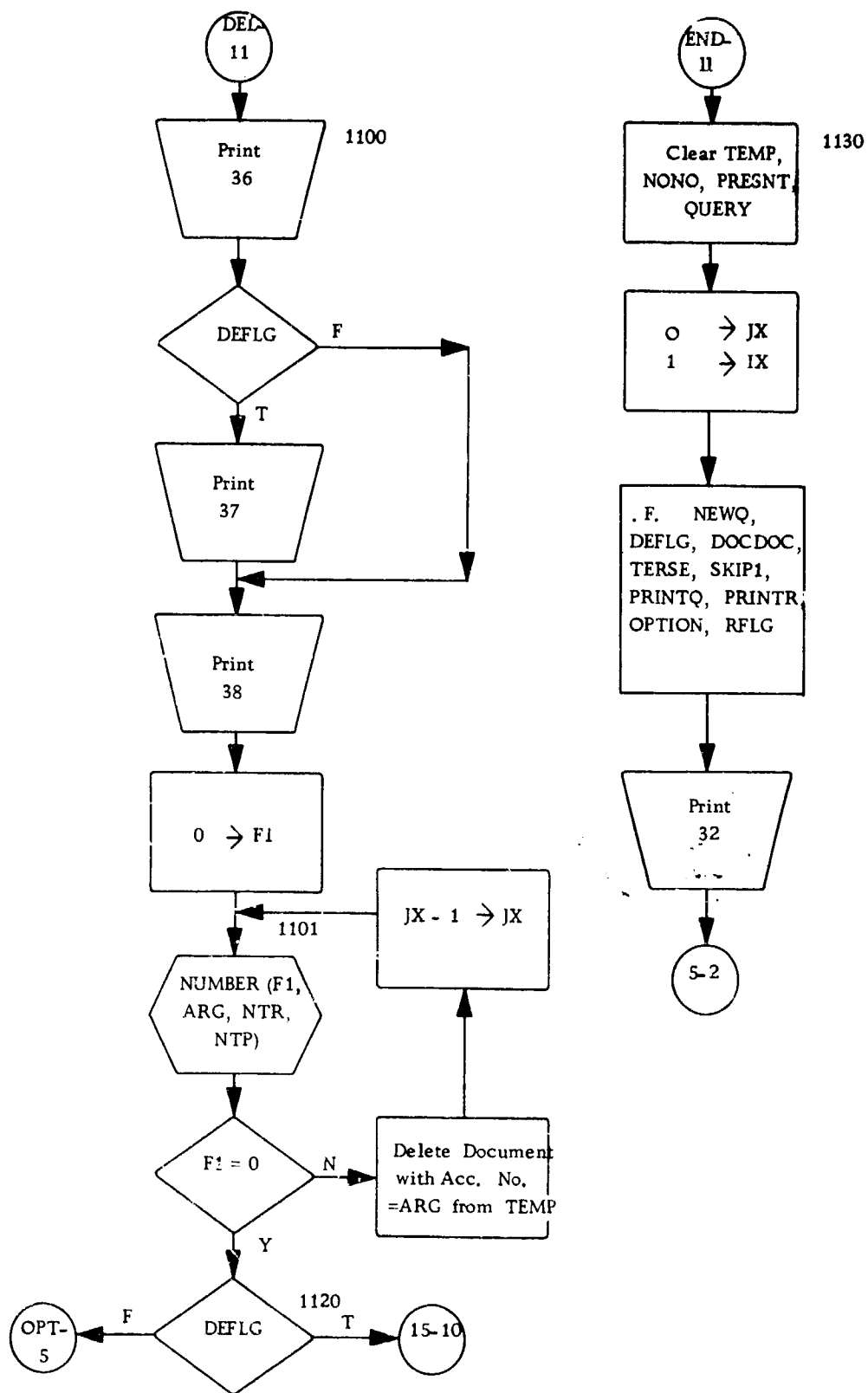
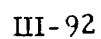


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR



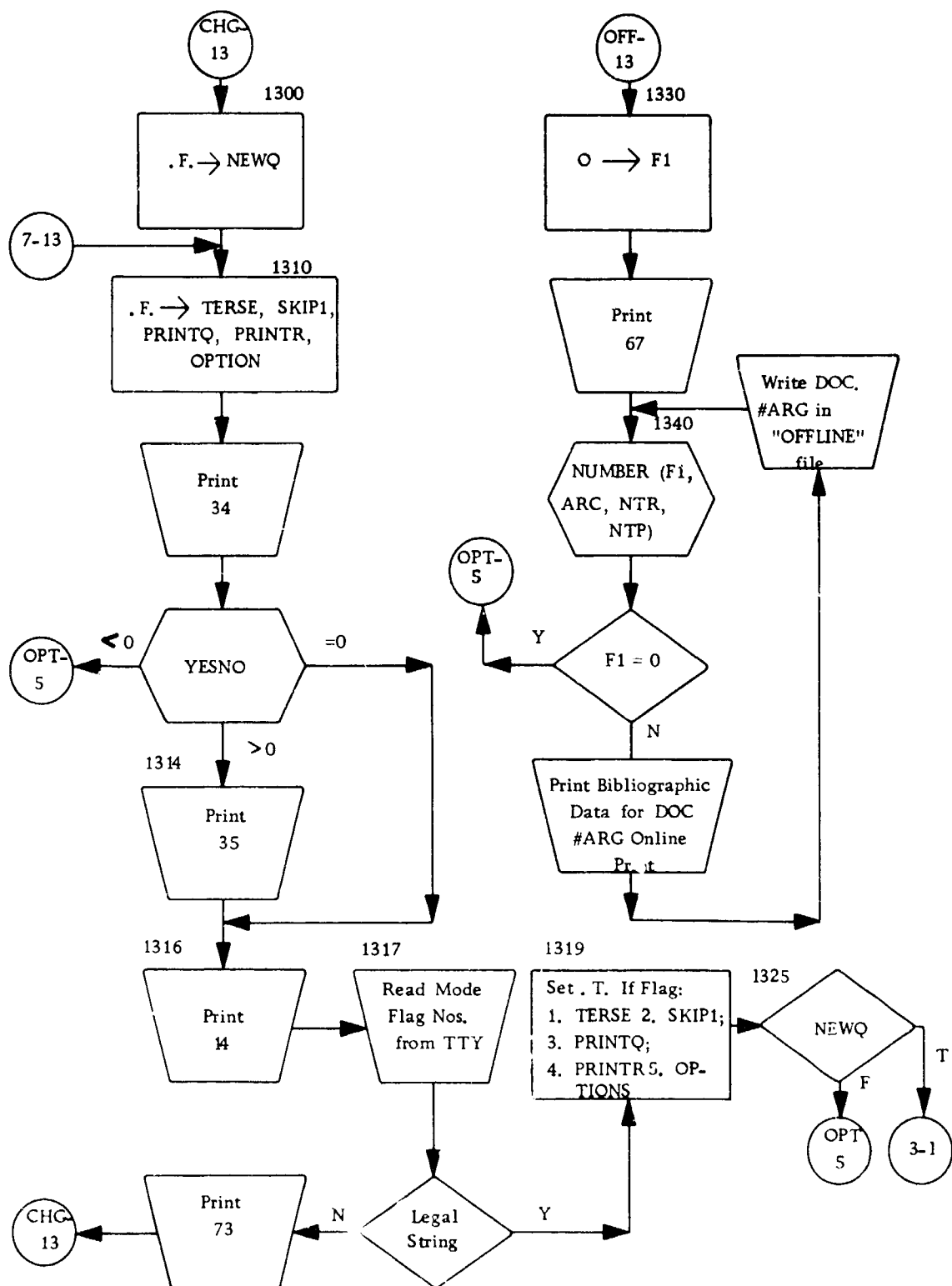


Figure III-13 (cont'd)  
DIALOGUE PROCESSOR

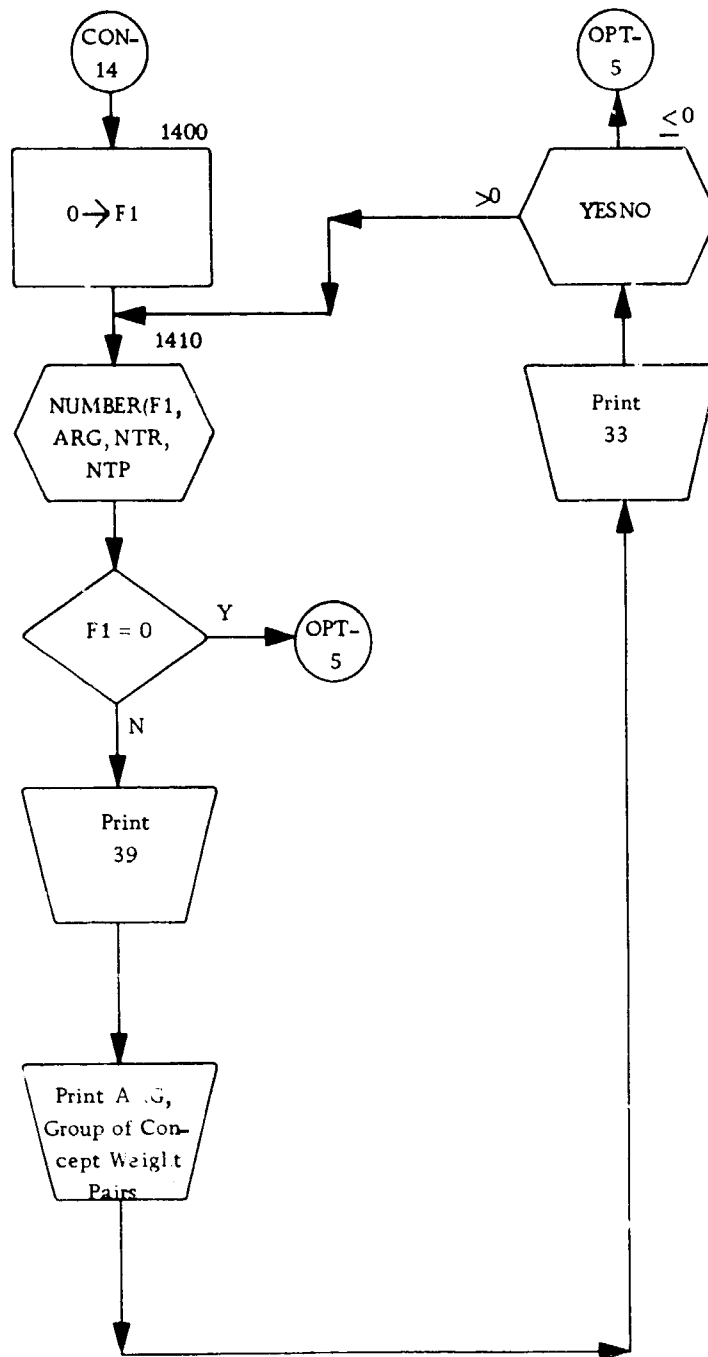


Figure III-13 (concluded)  
DIALOGUE PROCESSOR

## SECTION IV

### SUBROUTINES CALLED BY THE DIALOGUE PROCESSOR

This section briefly describes, and flowcharts, each subprogram called by the dialogue processor.

Figure IV-1 is a directory of all programs and subprograms that comprise the dialogue processor and the file-constructing programs. This directory tabulates the location within this report of the description and listing of each routine. Flowcharts are all co-located with descriptions.

#### IV.1 FUNCTION DOCK

##### IV.1.1 Purpose

To fetch authors' names, titles and abstracts from the test data file DATA1 selectively.

##### IV.1.2 Action

Upon a call to the logical function DOCK (CODE, I, ARRAY, COUNT), the following are input parameters:

CODE, an integer that is 1, 2 or 3 according as titles, authors or abstracts are desired.

I, an integer that specifies the accession number of the desired document.

Output parameters are:

ARRAY, eighteen words of ASCII information representing one line of the returned alphanumeric information.

COUNT, the number of words in ARRAY preceding the point (if any) where all remaining words are filled with blanks. This is used to avoid printing blanks that fill out lines.

	Calls	Called By	Described on page #	Listing on page #
<u>PROGRAMS</u>				
CONGRA	PLUCK, STEM		III-64	VI-41
DIALOGUE	All subroutines, directly or indirectly.		III-1	VI-22
DICGEN	PLUCK, STEM		III-55	VI-43
<u>SUBPROGRAMS</u>				
DOCK	None	DIALOGUE	IV-1	VI-2
LENGTH	None	STEM, NUM- BER	IV-3	VI-4
LOOKUP	None	DIALOGUE	IV-6	VI-7
NUMBER	LENGTH, OUT PLUCK, PUT, YESNO, ZORCH	DIALOGUE	IV-11	VI-10
OUT	None	DIALOGUE, NUMBER, YESNO	IV-19	VI-13
PLUCK	PUT	NUMBER, STEM, CONGRA, DICGEN, DIALOGUE	IV-19	VI-14
PUT	None	PLUCK, NUM- BER, STEM, CONGRA, DICGEN	IV-23	VI-16
STEM	LENGTH, PLUCK, PUT	DIALOGUE, CONGRA, DICGEN	IV-23	VI-17
WHERE	None	DIALOGUE	IV-28	VI-20
YESNO	OUT	DIALOGUE, NUMBER	IV-34	VI-21
ZORCH	None	NUMBER	IV-11	VI-10

Figure IV-1  
Directory of Programs and Subprograms

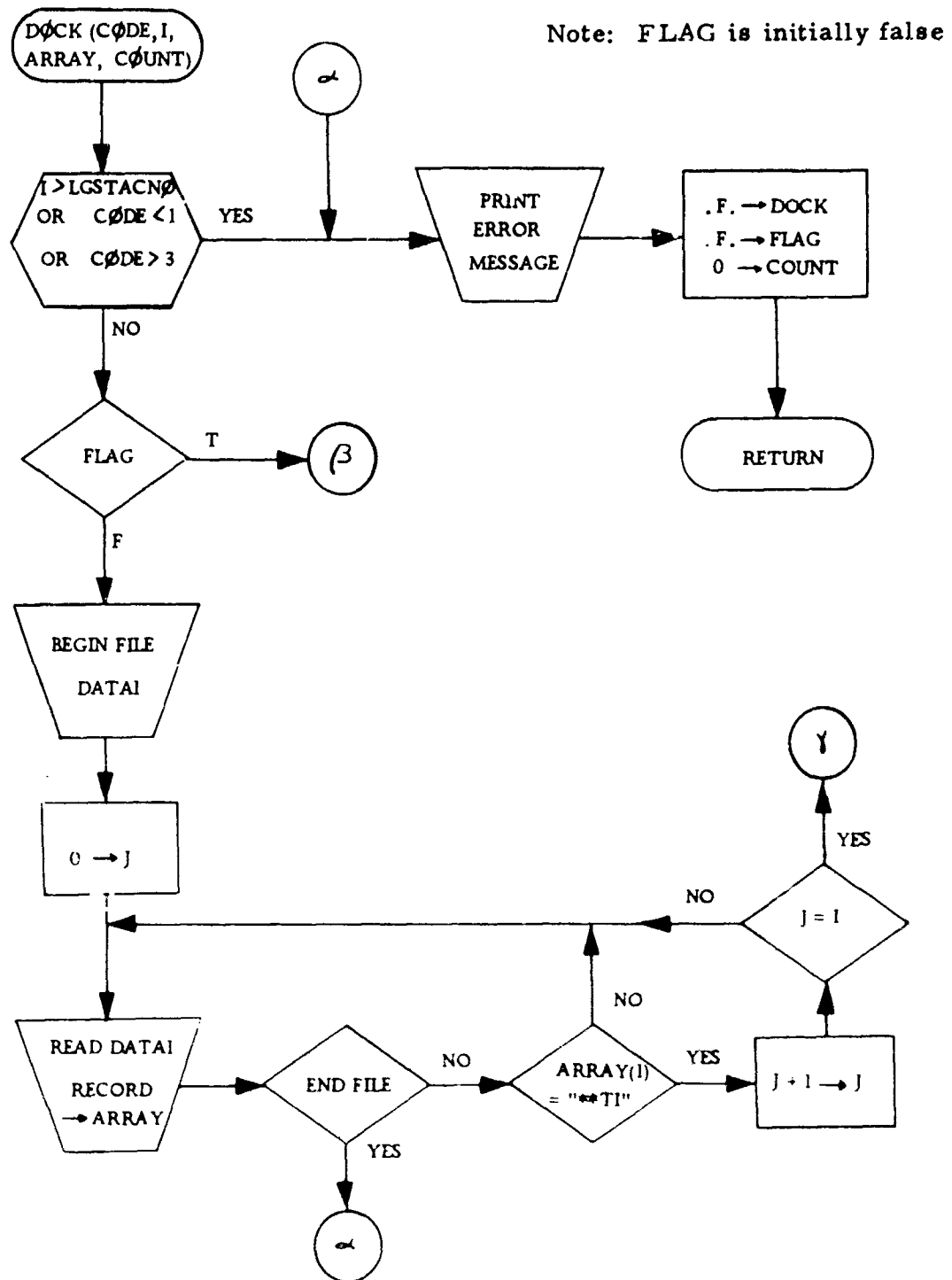


Figure IV-2 (cont'd)  
Function DOCK

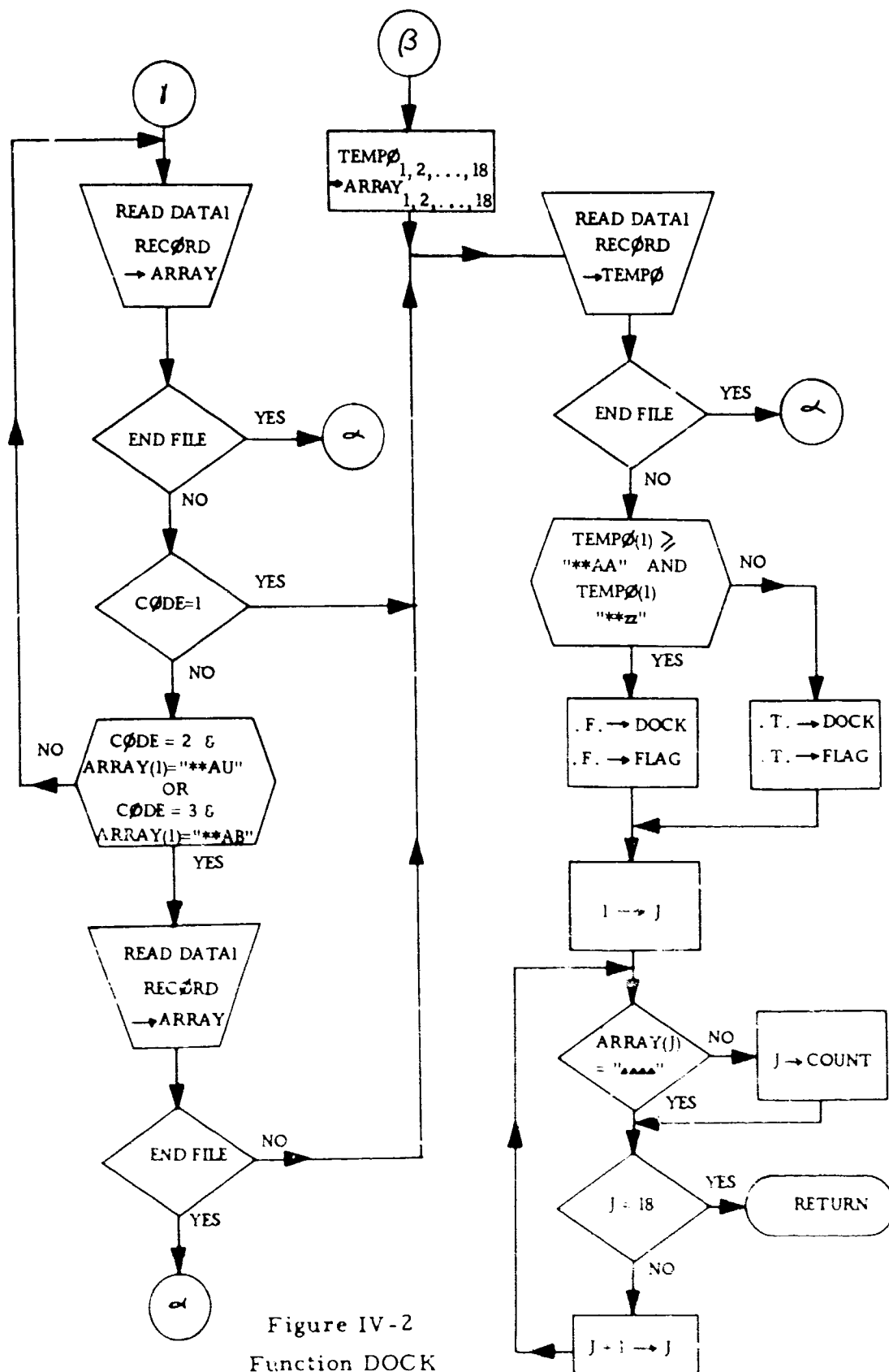


Figure IV-2  
Function DOCK  
(concluded)

DOCK, the logical value of the function. It is true if more lines of the requested data exist; subsequent calls to DOCK with the same input parameters will return additional lines of the data in ARRAY until the last line is delivered. When the last line has been transmitted, the value of DOCK will be false.

If CODE is different from 1, 2 or 3 or I is greater than the largest accession number, the routine will print an error message and return with DOCK false and COUNT=0. Recall that accession numbers entered by remote users pass through NUMBER, and that subprogram has the task of gracefully informing the user when he specifies an illegal accession number.

#### IV.1.3      Method

Data are read sequentially, with the first four characters of each line being scanned in order to determine the beginning of documents and fields within documents. Before a line is transmitted, the following line is checked to see if the transmitted line is the last of a sequence. If so, DOCK is made false. If more lines follow, the second line is saved for transmission on the next call.

### IV.2          FUNCTION LENGTH

#### IV.2.1      Purpose

Function LENGTH splits strings and counts their length

#### IV.2.2      Action

In a call to LENGTH (INPUT, RIGHT, LEFT, CUT), INPUT and CUT are the input parameters. All variables in the calling program are typed alphanumeric except for CUT and the value of the function, which are integers. The alphanumeric variables are stored in arrays of five words, left justified with remaining spaces filled out by blanks.

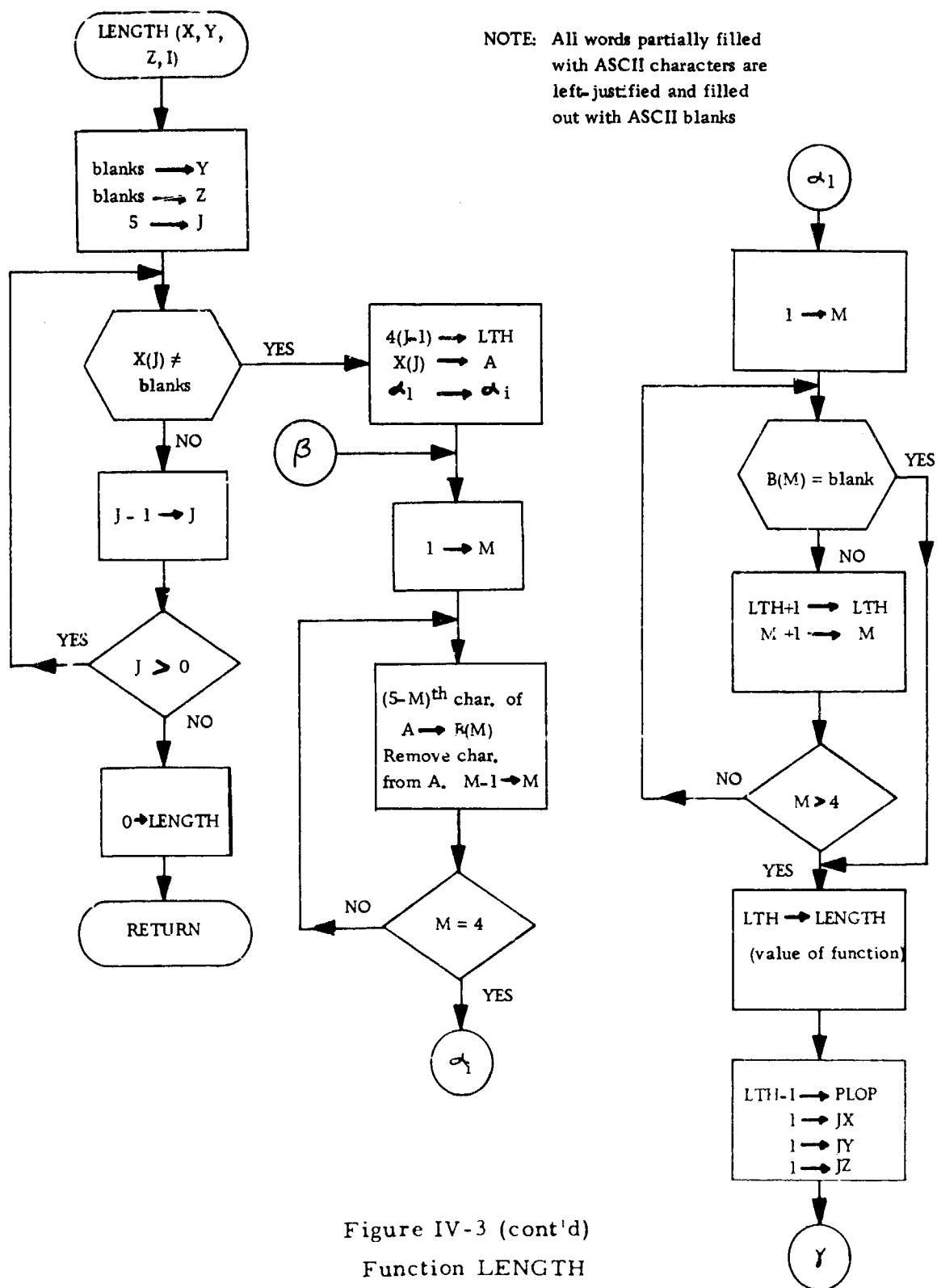


Figure IV-3 (cont'd)  
Function LENGTH

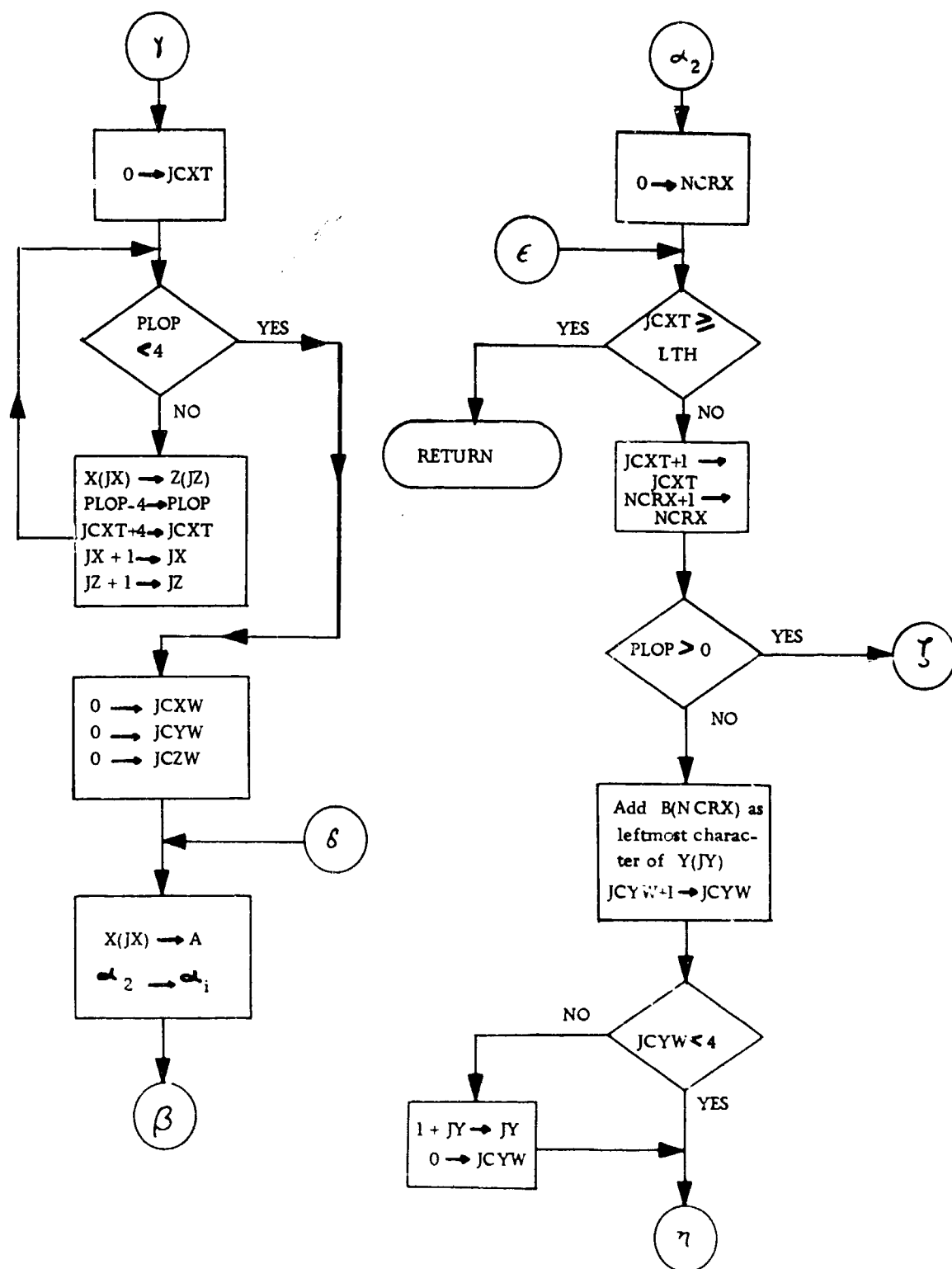


Figure IV-3 (cont'd)  
Function LENGTH

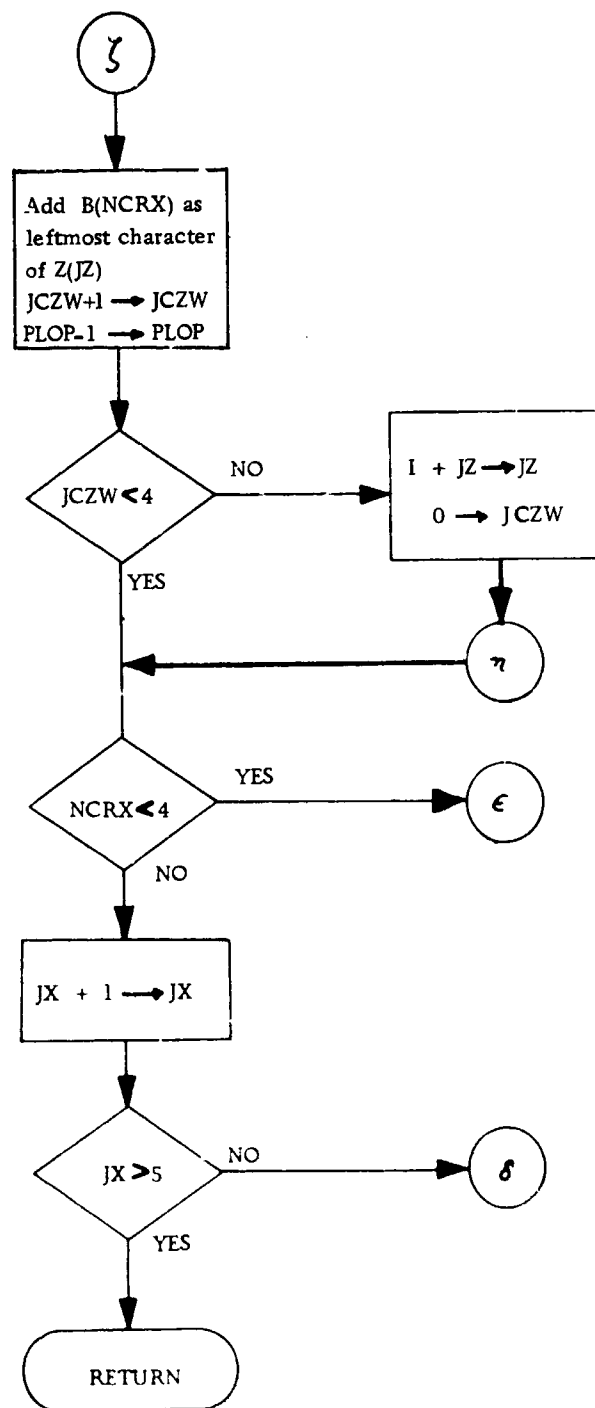


Figure IV-3 (concluded)  
Function LENGTH

LENGTH takes the input string and counts the number of characters in it, up to the first blank encountered or the end of a totally filled input array. The count is returned as the value of the function; suppose this is called L. The characters may be numbered 1, 2, . . . , L, with the first character in the input being number one. Upon return, characters 1, 2, . . . , L-CUT are returned in LEFT and characters CUT, CUT+1, . . . , L are in RIGHT. If CUT is greater than or equal to L, the entire string is returned in RIGHT; if CUT=0 the entire string goes to LEFT. Negative values of CUT are not allowed. INPUT is not altered by use of the function, and a call with INPUT null (all blanks) results in a return with both LEFT and RIGHT null and the value of the function equal to zero.

#### IV. 2. 3      Method

See the flowchart of LENGTH. Note that in order to obtain efficient operation, characters are moved by the word in the formation of LEFT to as great a degree as possible.

### IV. 3          FUNCTION LOOKUP

#### IV. 3. 1      Purpose

To search the concept dictionary file for a given stem, and to provide the concept vector for that stem if it is present in the dictionary.

#### IV. 3. 2      Action

The function LOOKUP(I) is logical in type. Its input parameters are the value of I and the stem located in the I<sup>th</sup> position of the Present Table, PRE2(1, I), PRE2(2, I) and PRE2(3, I). The function searches the concept dictionary file DICTNRY for the stem in the specified position of the Present Table, returning a value in the function name of FALSE if the stem is not in the dictionary. If the stem is found, a value of TRUE is returned

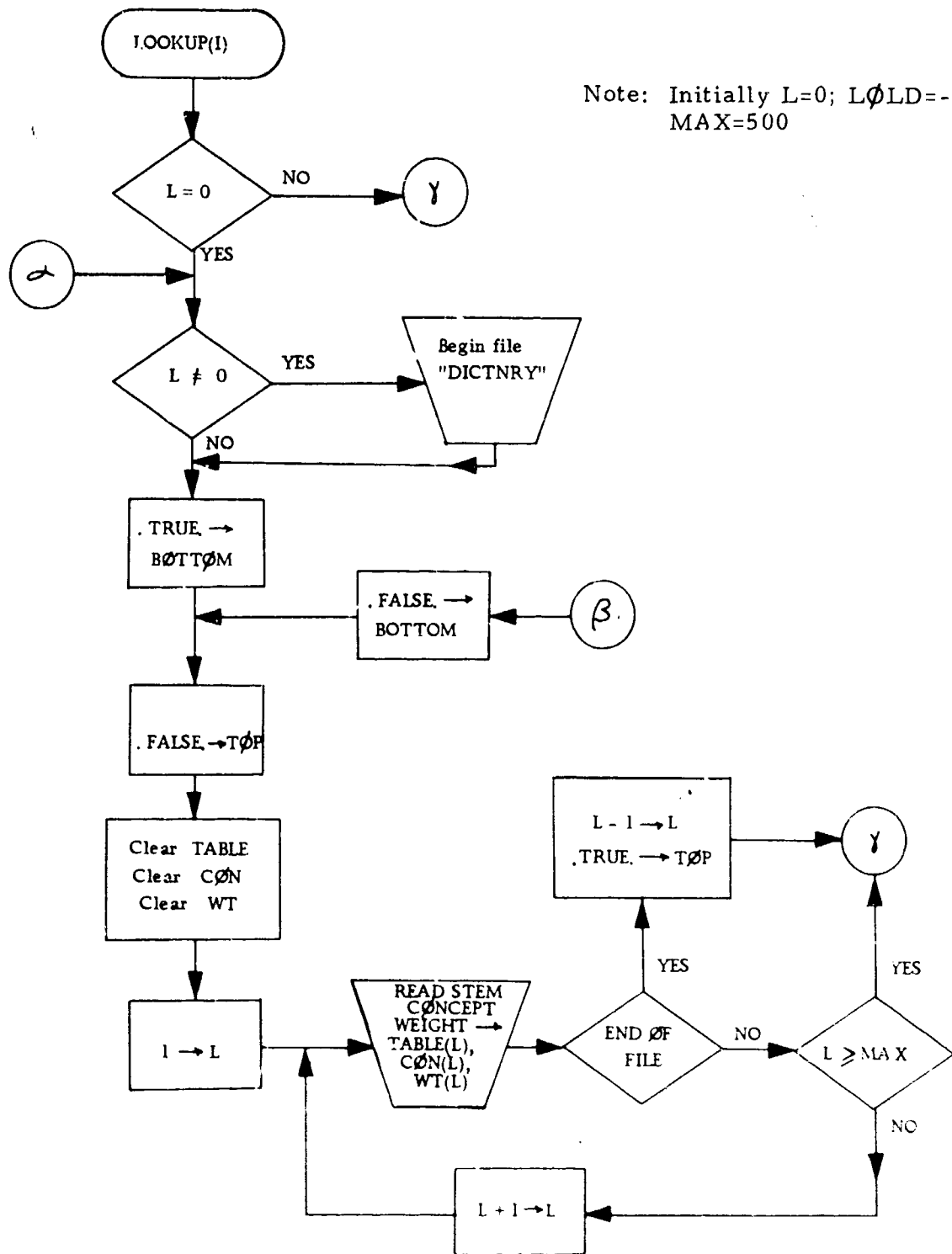


Figure IV-4 (cont'd)  
Function LOOKUP

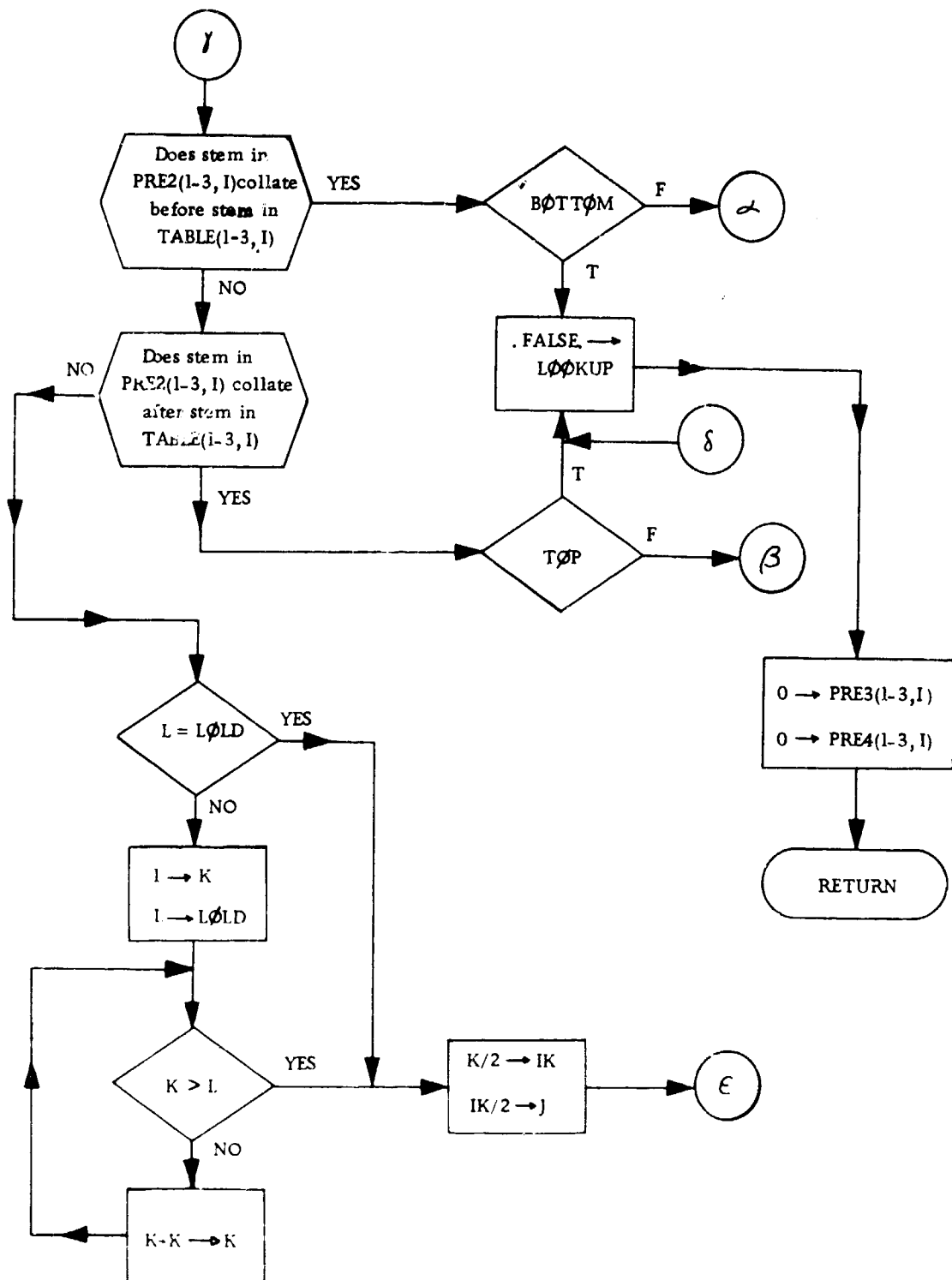


Figure IV-4 (cont'd)  
Function LOOKUP

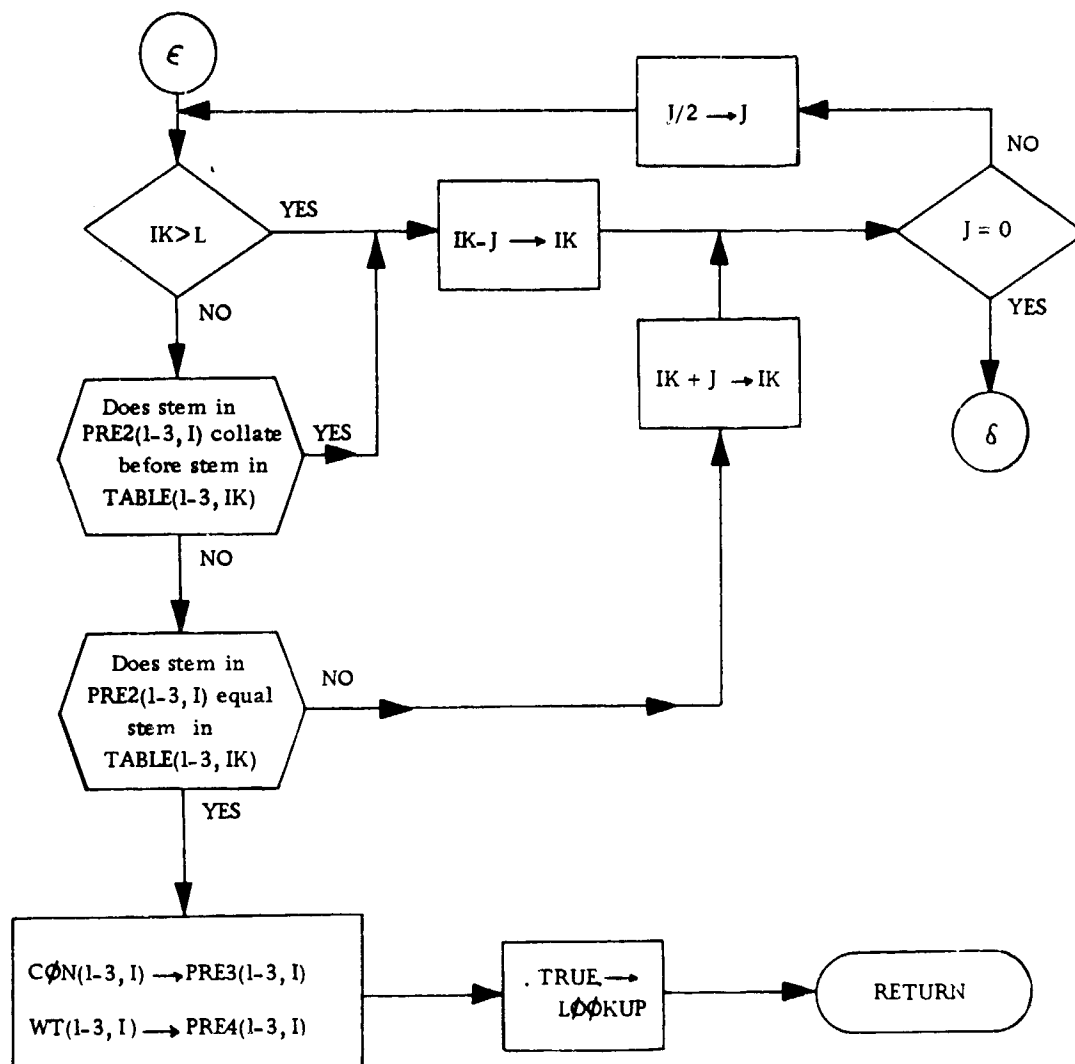


Figure IV-4 (concluded)  
Function LOOKUP

in the function name and concept-weight pairs are returned in the Present Table. The concept codes are placed in PRE3(1, I) through PRE3(3, I) and the corresponding weights are placed in PRE4(1, I) through PRE4(3, I).

In the event that a stem is not found in the dictionary, values of zero are returned for both the concept codes and the concept weights in the Present Table.

#### IV. 3. 3      Method

The dictionary lookup itself is performed using a binary search, and the length of the dictionary allowable is unlimited. This is accomplished by working with segments of the dictionary. The present segment length is 500 stems with their associated concept code-weight pairs, but this can be altered as available core permits.

When the routine is first entered, the status of the present dictionary segment is checked. If no segment is in core, then the first segment is read. There are flags which are set to indicate if the present segment is the first one (read immediately following a BEGIN FILE DICTNRY or on initial entry to LOOKUP), or the last (end of DICTNRY file encountered on last reading of a DICTNRY segment).

A stem is first checked against the lowest- and highest-collating stems of the dictionary segment presently in core. If it is outside of the limits, a higher or lower segment is read into core as appropriate. An exception occurs if the stem collates above the present segment and the present segment is the highest ordered one, or if the present segment is the lowest ordered and the stem collates below it. Then, clearly, the stem is not in the dictionary and so the associated concept code and weight positions in the Present Table are set to zero and the function returns with a FALSE value.

Once the correct segment is found, it is searched using a binary technique. It is necessary to establish a search starting point, and so first the smallest power of two not less than the number of entries in the present dictionary segment is computed. This step is omitted if the previous binary search operated on a segment of length equal to the present segment length; in practice all segments but the last one are of equal length owing to the characteristics of DICGEN, the dictionary generation program.

One-half of the value (smallest power of two not less than the size of the present segment) is used for the starting location of the search. After an unsuccessful comparison, a distance is either added to or subtracted from the starting location, according as the search found a stem above or below the desired stem. The distance is initially one-half of the starting location, and is of course halved after its application. If the distance is reduced to zero, the stem desired is not in the dictionary and so the corresponding concept code-weight pairs are set to zero and the function returns with a value of FALSE.

#### IV. 4        SUBROUTINE NUMBER (AND ASSOCIATED FUNCTION ZORCH)

##### IV. 4. 1    Purpose

As described in the On-Line Retrieval Interim Report (5), this routine reads document specifications from the remote terminal in a variety of forms, and returns document accession numbers and a status indicator.

##### IV. 4. 2    Action

The subroutine acts as described in the Interim Report, with three exceptions:

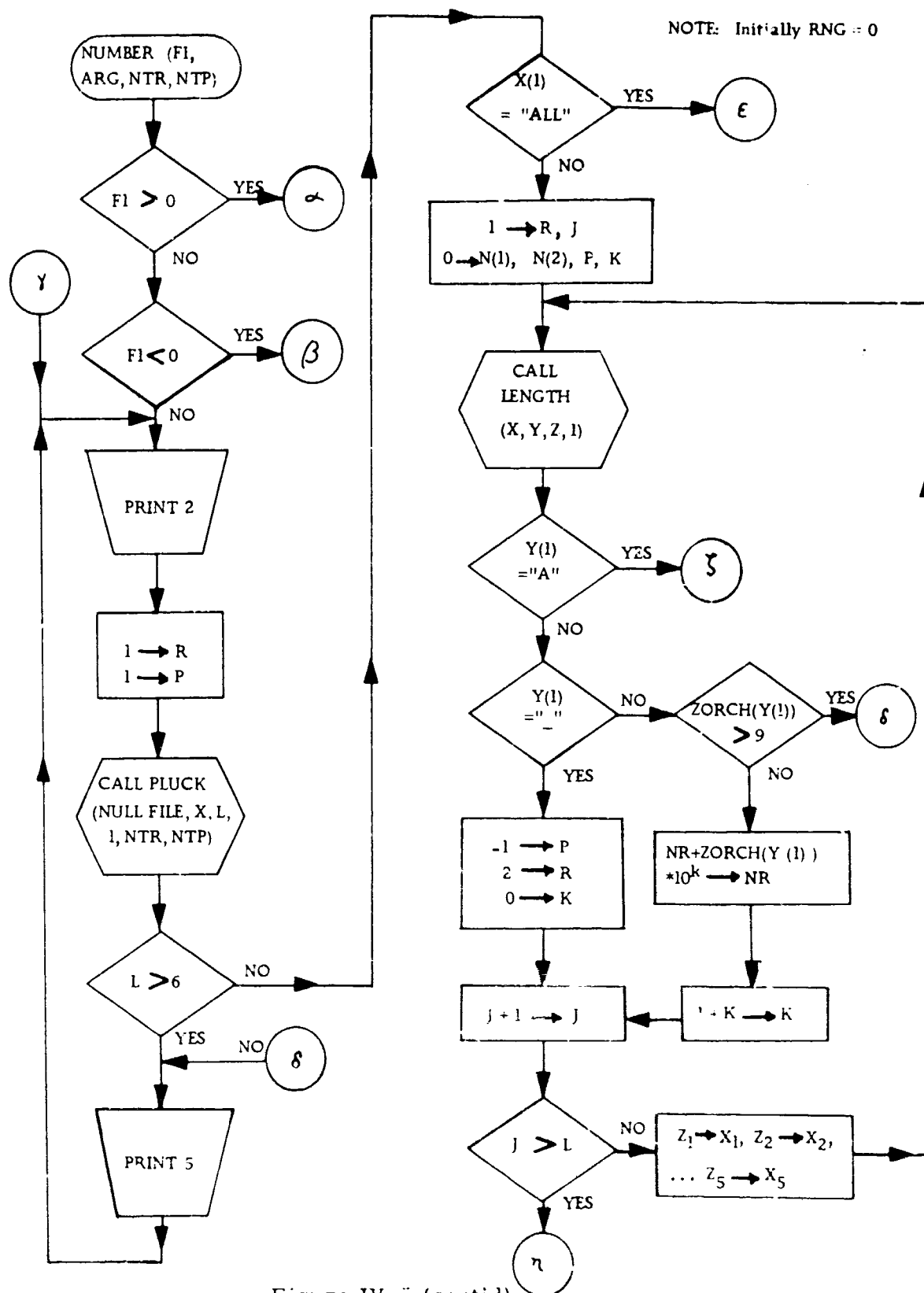


Figure IV-5 (cont'd)  
Subroutine NUMBER  
IV-15



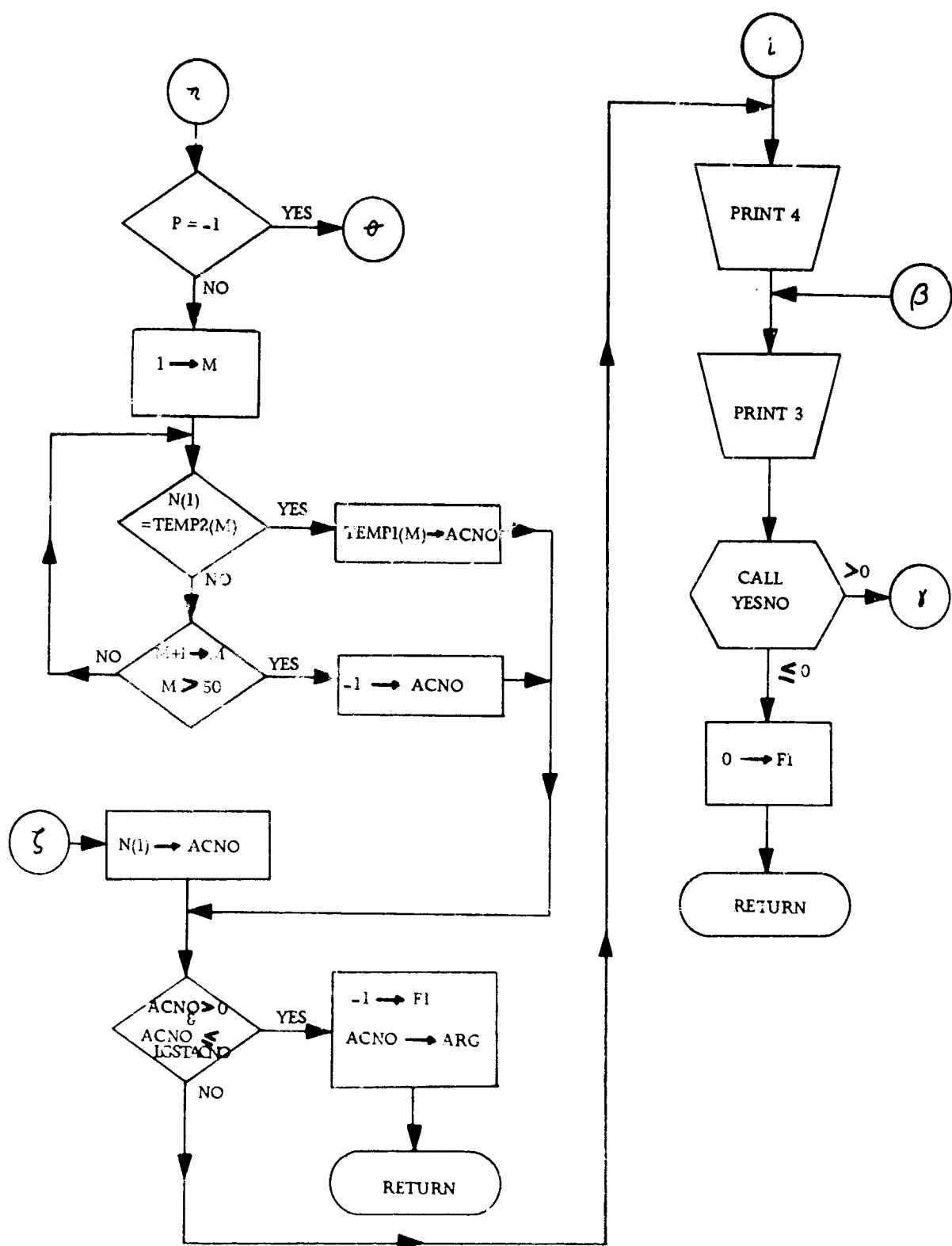


Figure IV-5 (concluded)  
Subroutine NUMBER  
IV-17

1. The Interim Report described a version of the routine which would print an error message at the remote console and request corrected input in the event that a non-existent document is specified. In the testing of NUMBER, it became apparent that this is not desirable in the case where a range specification of temporary document identification numbers is used. For example, a specification of all documents with temporary numbers in the range between thirteen and thirty ("13-30") would be reasonable even in the event that document number twenty-five had previously been deleted from the temporary file. The user should not be forced to specify "13-24" and "26-30". The routine accepts such range specifications even though not all documents exist. However, it will not accept a specification if none of the specified documents are present in the temporary file.
2. Leading zeros do not have to be provided with accession number specifications: "A17," "A017" and "A00017" are all accepted as identifying the document whose accession number is seventeen.
3. When the user is asked if he wishes to specify additional documents, a reply of "OPTIONS" is treated the same way as a reply of "NO".

Leading and trailing blanks are allowed in the specification.

#### IV. 4.3 Method

The subroutine uses PUCK to read delimited strings from the remote console, as that may be treated as a file. YESNO is used when the user is asked if more documents are to be specified; all messages are printed by means of calls to OUT. LENGTH and PUT are used by PUCK, and LENGTH is also called directly when the input specification strings are analyzed.

In order to avoid conversion problems, the transformation from ASCII to internal integer representation is programmed directly, rather than achieved by use of the ENCODE/DECODE statements. For this same reason, a small function ZORCH is used so that ASCII characters may be handled as integers. Since ZORCH is required for this reason, it is convenient to include in it a detection of non-numeric characters.

Except as noted above, the logic of NUMBER follows the description in the Interim Report.

## IV.5 SUBROUTINE OUT

### IV.5.1 Purpose

To print standard messages at the remote terminal.

### IV.5.2 Action

A call to OUT(J) causes standard message number J to be printed at the remote terminal. Recall that some messages exist in both terse and verbose forms. A logical parameter in common, TERSE, is true if terse dialogue is desired. The verbose form of the message is printed unless both TERSE is true and message number J possesses a terse form. In the event that OUT(J) is called with a value of J corresponding to no message number, the following error message is printed and control returned to the calling program:

ERROR IN 'OUT' SUBROUTINE AT MESSAGE # nn

where "nn" is the invalid value of J used in the call.

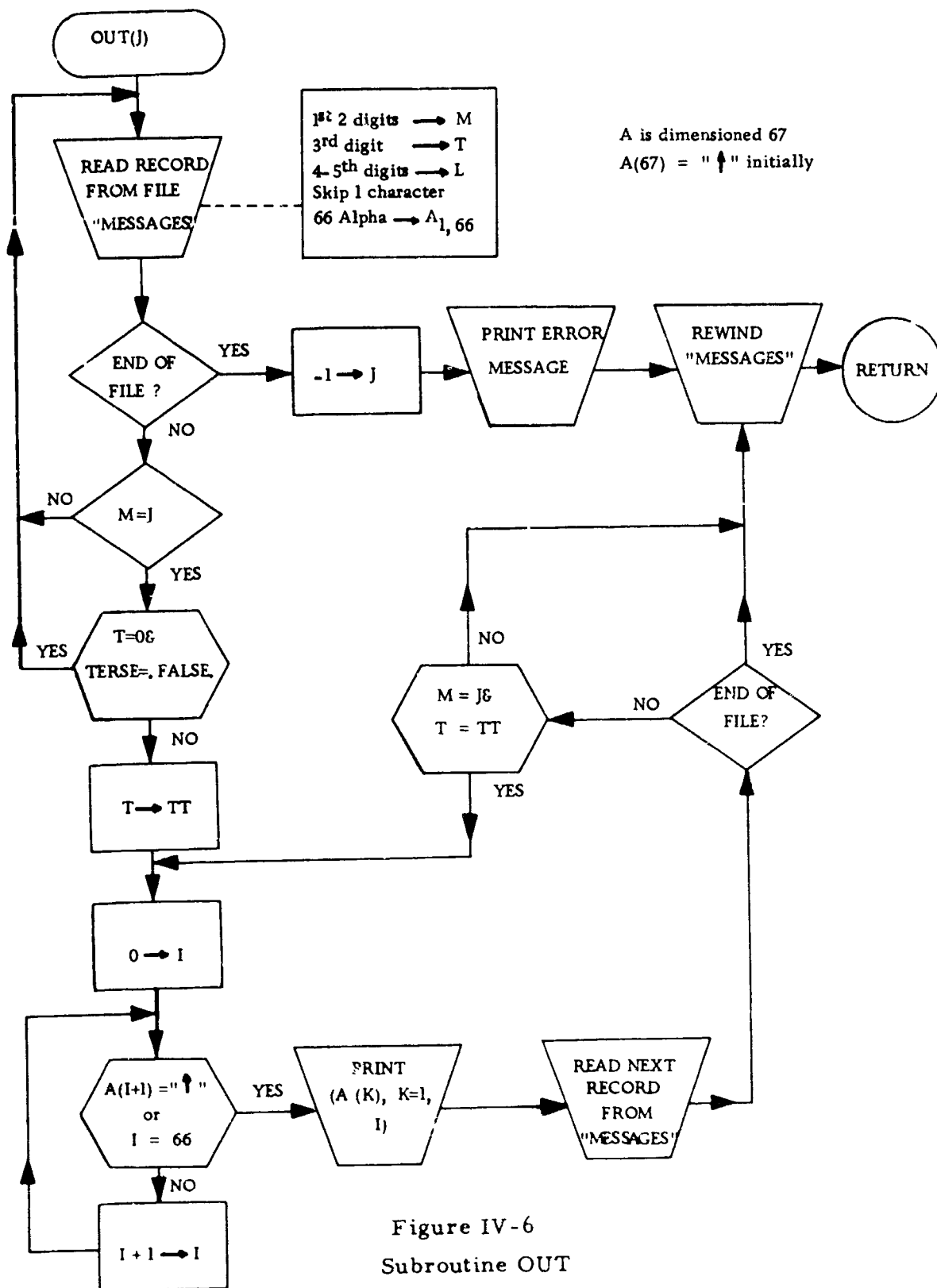
### IV.5.3 Method

For a description of the format of file MESSAGES, see section III.2.5.

## IV.6 SUBROUTINE PLUCK

### IV.6.1 Purpose

Subroutine PLUCK scans text files and returns character strings. Input parameters to PLUCK determine the delimiters used in string definition, the file to be searched and the starting point in that file. Output parameters are the string found, its length, and the position in the file of the start of the next scan. The position data may be saved in order to resume searching a



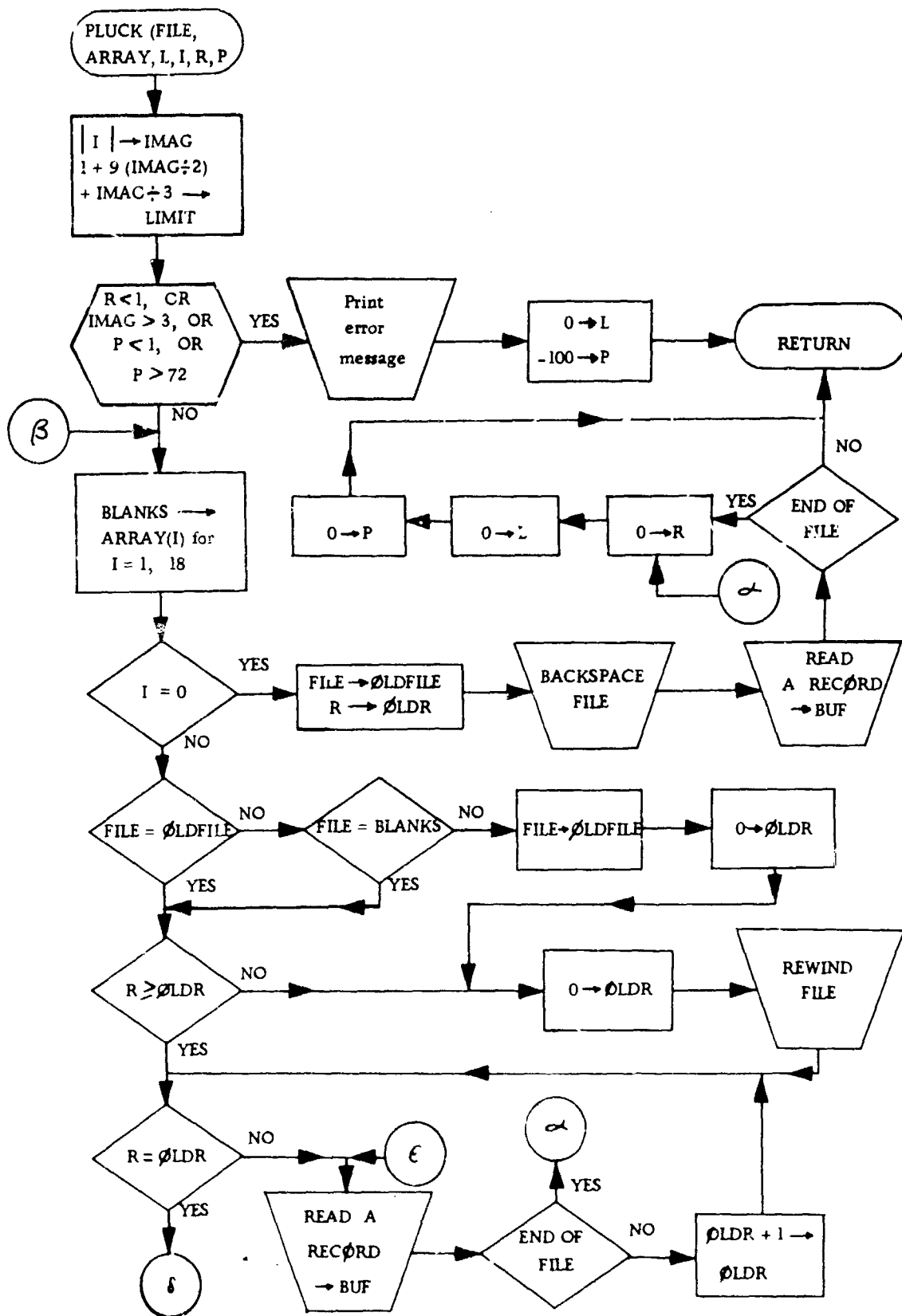


Figure IV-7 (cont'd)

Subroutine PLUCK

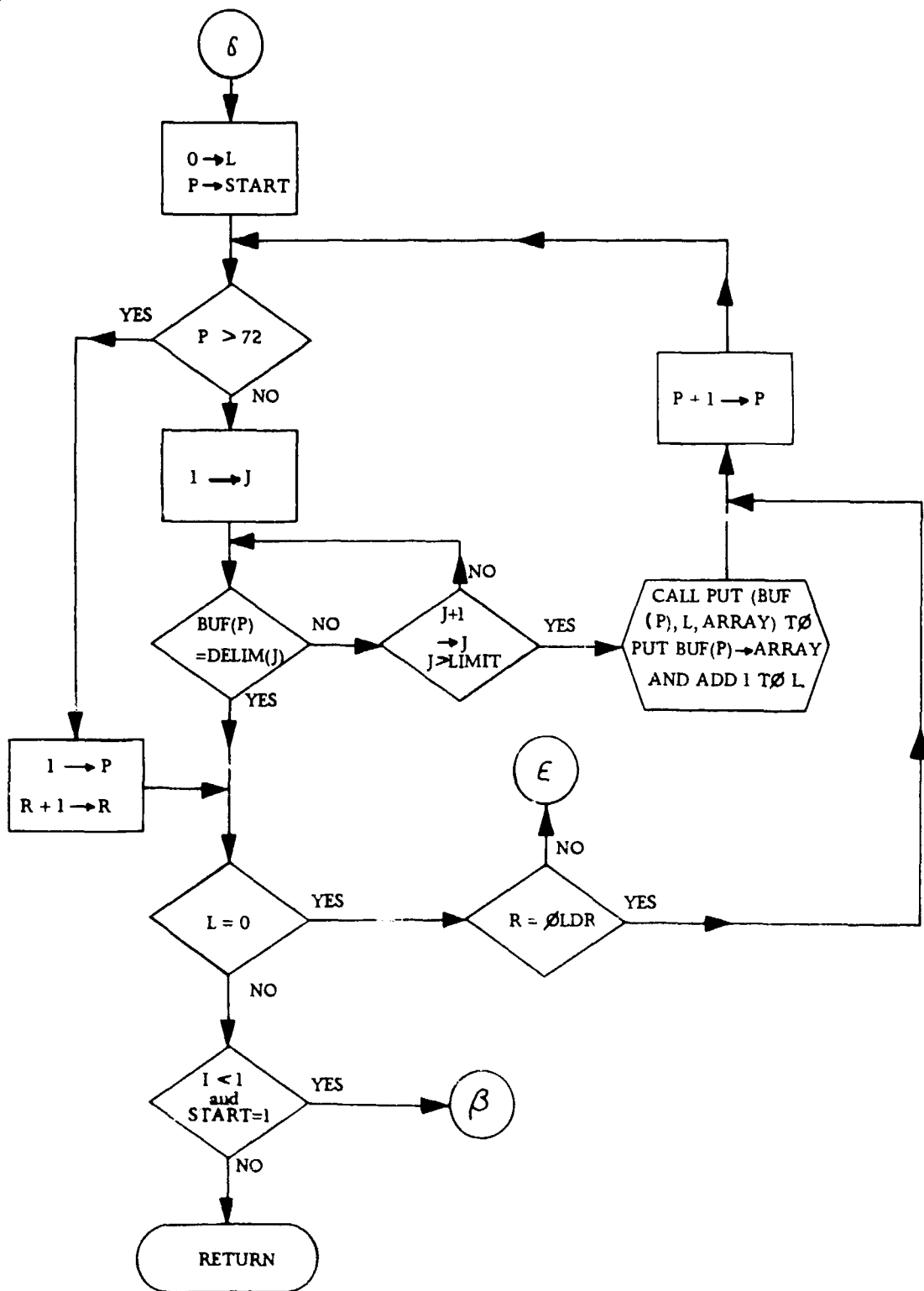


Figure IV-7 (concluded)  
Subroutine PLUCK

Output from file CHICKEN

I	L	R	P	ARRAY
1	8	17	13	JØURNALS
1	3	17	17	AND
1	5	17	23	TRADE
1	9	17	33	MAGAZINES
1	4	17	38	WERE
1	10	17	49	TØP-RANKED
1	2	17	52	AS
1	4	17	57	MØST
1	10	17	68	IMPØRTANT.
1	3	18	4	211
2	8	17	13	JØURNALS
2	3	17	17	AND
2	5	17	23	TRADE
2	9	17	33	MAGAZINES
2	4	17	38	WERE
2	10	17	49	TØP-RANKED
2	2	17	52	AS
2	4	17	57	MØST
2	9	17	67	IMPØRTANT
2	3	18	4	211
3	8	17	13	JØURNALS
3	3	17	17	AND
3	5	17	23	TRADE
3	9	17	33	MAGAZINES
3	4	17	38	WERE
3	3	17	42	TØP
3	6	17	49	RANKED
3	2	17	52	AS
3	4	17	57	MØST
3	9	17	67	IMPØRTANT
-3	8	17	13	JØURNALS
-3	3	17	17	AND
-3	5	17	23	TRADE
-3	9	17	33	MAGAZINES
-3	4	17	38	WERE
-3	3	17	42	TØP
-3	6	17	49	RANKED
-3	2	17	52	AS
-3	4	17	57	MØST
-3	9	17	67	IMPØRTANT

Figure IV-8  
Demonstration of PLUCK

file following a search of another file. If it is known that no intermediate activity will change the status of the first file, a special call to PLUCK can be made in order to avoid initialization after a switch from one file to another and back to the first.

In order to illustrate the workings of PLUCK, its listing here includes a test driver that reads from files CHICKEN and LENGTHFN (which contains function LENGTH). The driver contains comment lines that indicate the purpose of the calls to PLUCK, and Figure IV-8 shows the output obtained. More extensive tests were performed than those shown here.

#### IV.6.2     Action

In a call to PLUCK (FILE, ARRAY, L, I, R, P), the parameters have the following meaning:

FILE is a filename constant or variable, indicating the name of the file to be read. It is not required that the file be line-numbered. Unless either the last call to PLUCK obtained data from the same file or contained information on the file's status (see the case where I=0, below), the file will be rewound and repositioned upon a call.

ARRAY contains the string found, up to 72 characters left-justified in ASCII format and filled out with blanks.

L contains the number of characters in the string found. If the end of the file is reached L=0.

R is both an input and output parameter. In a call it indicates the sequential number (starting at one) of the first record to be searched. On output it indicates the number of the record containing the string found, or the next record if the string found was the last one in a record. If the end of file is reached, R=0.

P is like R, except that it indicates the next character position to be searched within the record. If the end of file is reached, P=0. If one of the calling parameters is outside the legal limits, P=-100. It is assumed that the maximum length of a record is 72 characters, but this constraint is easily changed.

I is the input parameter which controls the action of PLUCK in the selection of delimiters and the control of file initialization.

I=0. This call does not return a string. It is used when switching from the file last referenced back to a file referenced previously where the last value of R is known. It backspaces the previous file one record, and reads that record so that any following calls to that file specifying a record number R or greater do not require a complete file rewind and reread to record R.

I=1. The string starting with or following the character number P in record R is obtained. The only recognized delimiters are the space and end of record.

I=2. Like the case with I=1, except that comma, period, double quote, exclamation point, colon, semicolon, right and left parenthesis and question mark are recognized as delimiters in addition to space.

I=3. Like I=2, except that the dash is included as a delimiter.

I=-1. Like I=1, except that the first string of a record is ignored. This is for use with line-numbered files.

I=-2; I=-3. Like I=2 and I=3, respectively, except for line-numbered files.

#### IV. 6. 3      Method

Figure IV-7 shows a flowchart for subroutine PLUCK; Figure IV-8 is a listing of the program and a demonstration driver.

The use of the variables FILE 1 and FILE 2 is not obvious. Certain versions of the compiler have shown errors in the handling of IF statements dealing with filename variables, but all versions allow the

replacement statement to have filename variables on the right and ASCII variables on the left. If difficulties are encountered with the IF statement, then FILE 1 and FILE 2 may be declared ASCII rather than FILENAME. This will overcome the compiler difficulties, but it must be remembered that then only the leftmost four characters of the file names will be compared. PLUCK calls on the function PUT, described below.

#### IV.7        FUNCTION PUT

##### IV.7.1     Purpose

Function PUT is used by PLUCK to pack single characters into words. Since it provides a generally useful capability, it has been written as an integer function rather than as part of Function PLUCK.

##### IV.7.2     Action

Integer function PUT has three parameters: a 72 character array ARRAY, an integer L and a single character A. A call to PUT (A, L, ARRAY) puts character A into position L+1 of the 18 word array ARRAY, and returns a function value of L+1. The character in A must be left-justified and filled with blanks, and ARRAY must similarly contain blanks.

#### IV.8        SUBROUTINE STEM

##### IV.8.1     Purpose

Subroutine STEM performs stem analysis and common word detection.

The degree of suffix removal is governed by a single constant (Line 79750) of STEM. The desirable value of this constant was anticipated to lie between four and six, as stated in the Interim Report. Five has been

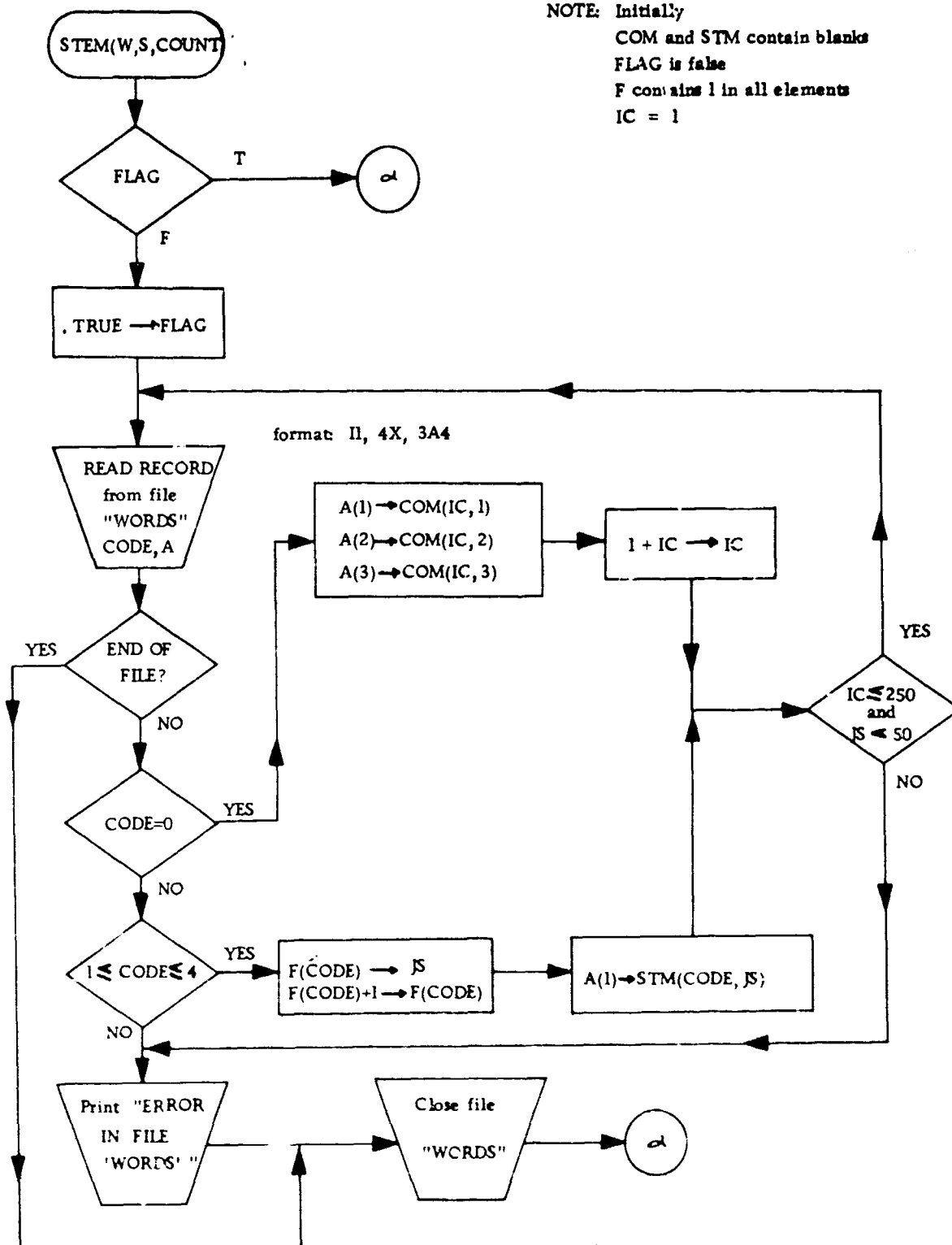


Figure IV-9 (cont'd)  
Subroutine STEM

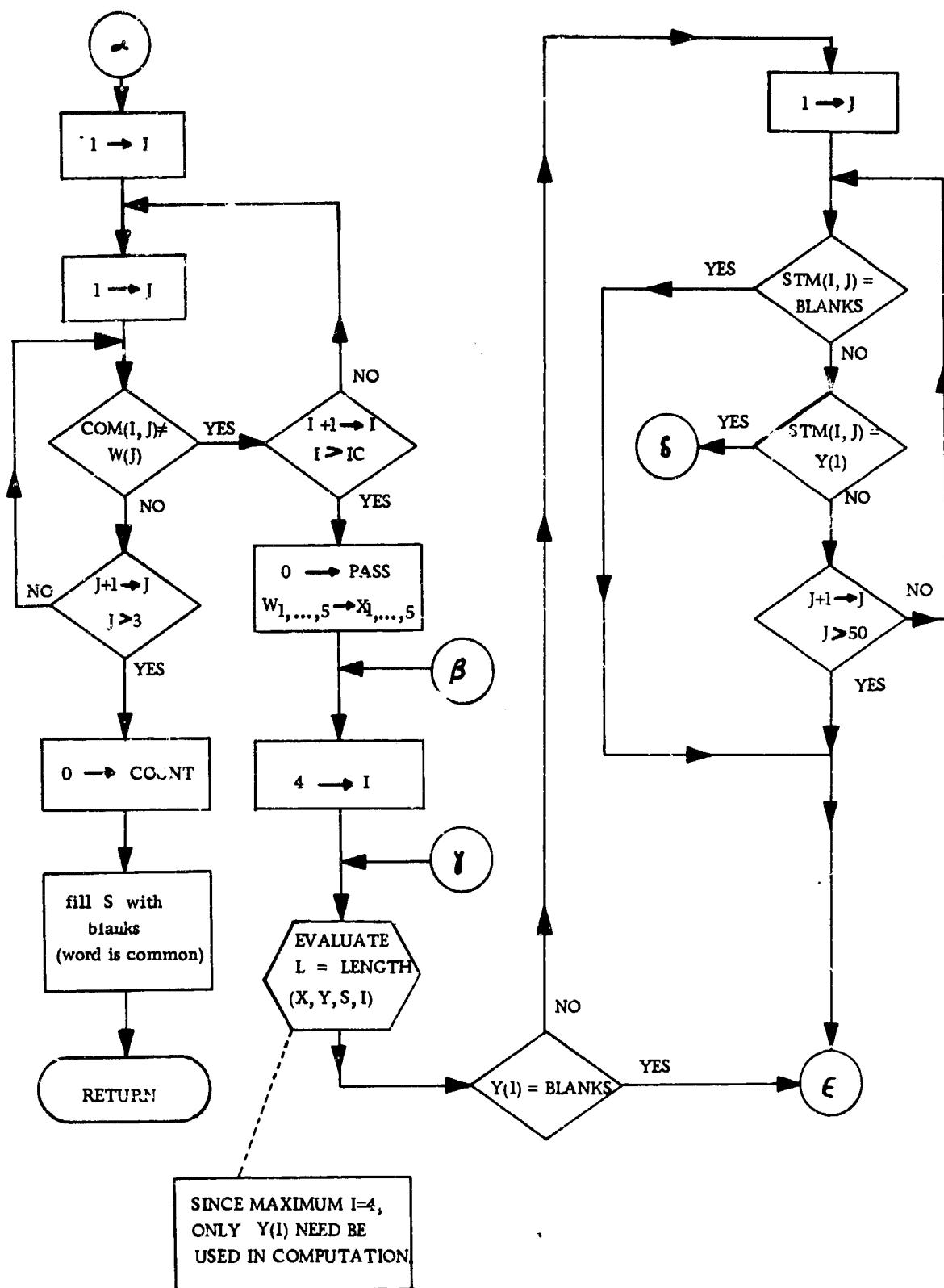


Figure IV-9 (cont'd)  
Subroutine STEM

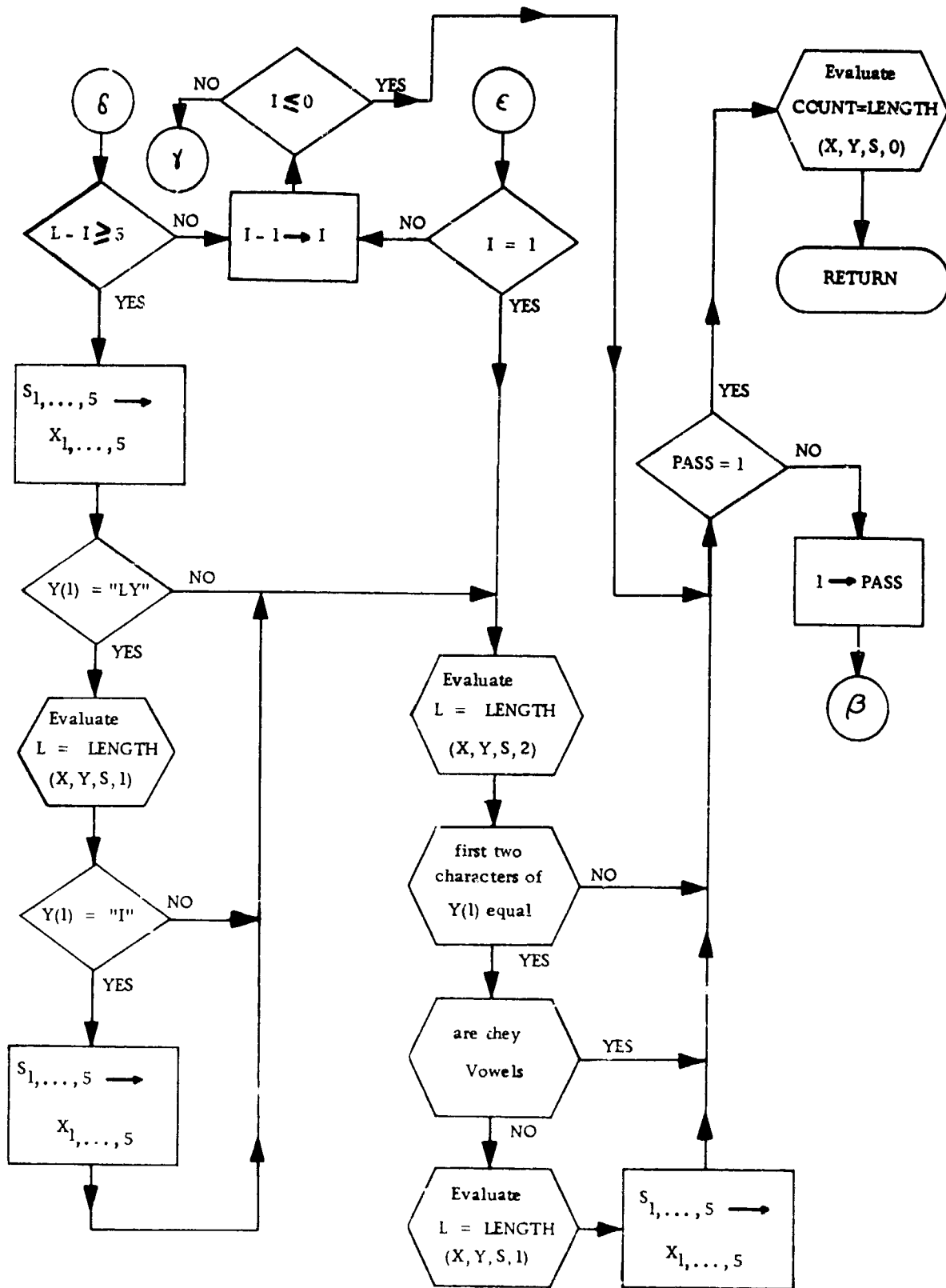


Figure IV-9 (concluded)  
Subroutine STEM

selected for the present time, but of course it can be revised in accordance with future experience. Note that stems of fewer than five characters can be generated by the special cases which remove a terminal "i" after ordinary removal of "ly" and which remove the second of a double terminal consonant. For a discussion of this and other aspects of the performance of STEM, see subsection III.2.3 of this report.

#### IV.8.2     Action

STEM (WORD, STEMRD, COUNT) has a single input, an array of up to 20 characters named WORD. The subroutine returns with WORD unchanged, the stem found in STEMRD (another array of up to 20 characters) and the length of the stem in COUNT. If the input word is common, STEMRD contains blanks and COUNT=0.

STEM requires a file WORDS, in the format described above, containing up to 250 common words and up to 50 suffixes each of length one, two, three and four. If a format error is found in that file, an error message is printed. Processing continues using the part of the file read. Of course, certain invalid file data can cause a TSS system abort, over which the Dialogue Processor has no control.

#### IV.8.3     Method

The common words and stems are stored in a separate file, WORDS. This is done in order that they may be modified without altering and/or recompiling STEM. This file is read and stored by STEM upon the first call to STEM, and then the file is closed.

Stem analysis is performed as described in the Interim Report.

#### IV. 8. 4      Example

A short program was written to read from file CHICKEN, with contents as shown in Figure IV-10(a), using subroutine PLUCK and placing the results in an eighteen word array FEATHERS. STEM was called with FEATHERS as its argument, resulting in the output shown in Figure IV-10(b). Each asterisk indicates a rejected common word.

The control parameter of PLUCK was set to +3, and functions LENGTH and PUT were of course also loaded. It should be noted that the use of FEATHERS as output of PLUCK and input to STEM is perfectly permissible, even though the dimensioning statements within those subroutines are different.

### IV. 9          SUBROUTINE WHERE

#### IV. 9. 1      Purpose and Action

In the debugging of a complicated program such as the Dialogue Processor, the programmer is frequently faced with the problem of determining the path of control through the program. This subroutine is designed to aid in that determination. Calls are of the form CALL WHERE (A, N), where A is an ASCII constant and N is an integer. The subroutine responds by printing the values of A and N. Successive calls produce printing of A and N on one line, until that line is filled then a new line is started. There are two exceptions to this:

1. As debugging progresses, the programmer may wish to turn off the action of WHERE. Therefore, when it is first called, it prints "ACTIVATE TRACE?" An answer of "NO" will suppress all printing by WHERE; when called it will immediately transfer control back to the calling program.

100 THE SURVEY SHOWS THAT, WHILE THE RESULTS ARE NOT INEXPLICABLY  
101 CONTRADICTORY, DIFFERENCES IN PRINCIPLE AND METHOD MAKE IT IMPOSSIBLE  
102 TO DEMONSTRATE CERTAIN, CLOSE AGREEMENT. THE AUTHOR SUGGESTS THAT  
103 FUTURE SURVEYS SHOULD BE DESIGNED TO INCLUDE A FEW FEATURES WITH A  
104 DELIBERATE RELATIONSHIP TO EARLIER SURVEYS SO THAT SOME VALID  
105 COMPARISONS CAN BE MADE.  
200 IN SURVEY ON THE USE OF LITERATURE INVOLVING 1800 ENGINEERS,  
201 300 WERE PERSONALLY INTERVIEWED. THE SAMPLE CONSISTED OF 206  
202 CHEMICAL ENGINEERS AND THE BALANCE (1594) OTHER ENGINEERS NOT  
203 SPECIFIED. THE + SURVEY SHOWED THAT 85 PERCENT OF THE CHEMICAL  
204 ENGINEERS, AND 85 PERCENT OF OTHER ENGINEERS, USED MANUFACTURERS'  
205 CATALOGUES IN THEIR WORK. OTHER TYPES OF LITERATURE SURVEYED AND  
206 OTHER ENGINEERS USING THEM ARE AS FOLLOWS: REPRINTS, 73, 70;  
207 PREPRINTS, 50, 37; ABSTRACTS AND INDICES, 51, 33; HANDBOOKS,  
208 TRANSLATIONS, 35, 22; TELEVISION IN RELATION TO WORK, 5, 6;  
209 STANDARDS AND SPECIFICATIONS, 49, 63; PATENTS, 54, 24. SIXTEEN  
210 JOURNALS AND TRADE MAGAZINES WERE TOP-RANKED AS MOST IMPORTANT.  
211 MANUFACTURERS CATALOGUES, ABSTRACTS AND INDICES, AND PATENTS  
212 WERE ALSO CATEGORIZED BY PERCENT OF USE BY FUNCTION GROUPS OF  
213 ENGINEERS (CHEMICAL, CONSULTING, DESIGN AND DEVELOPMENT, FACILITIES  
214 PLANNING, AND INFORMATION RETRIEVAL).

Figure IV-10(a)  
File CHICKEN

10 \* SURVE SHØWS \* \* \* RESULT \* \* INEXPLICAB  
 101 CONTRADICTØ DIFFER \* PRINCIPAL \* METHOD MAKE \*  
 IMPØSSIBL 102 \* DEMØNSTR CERTAIN CLØSE AGREE \* AUTHØR SUGGEST \*  
 103 FUTUR SURVE \* \* DESIGN \* INCLUD \* \* FEATUR \* \*  
 104 DELIB RELAT \* EARLI SURVE \* \* \* VALID  
 105 CØMPARIS \* \* MADE  
 20 \* SURVE \* \* USE \* LITERAT INVØLV 180 ENGINE  
 201 30 \* PERSØN INTERVIEW \* SAMPL CØNSIST \* 206  
 202 CHEMIC ENGINE \* \* BALANC 1594 \* ENGINE \*  
 203 SPECIF \* + SURVE SHØWED \* 85 PERCENT \* \* CHEMIC  
 204 ENGINE \* 85 PERCENT \* \* ENGINE USED MANUFACTUR  
 205 CATALØGU \* \* WØRK \* TYPES \* LITERAT SURVE \*  
 206 \* ENGINE USING \* \* \* FØLLØW REPRINT 73 70  
 207 PREPRINT 50 37 ABSTRACT \* INDIC 51 3 HANDEØØK  
 208 TRANSLA 35 2 TELEVI \* RELAT \* WØRK 5 6  
 209 STANDARD \* SPECIFICA 49 63 PATENT 54 24 SIXTE  
 210 JØURN \* TRADE MAGAZIN \* TØP RANKED \* \* IMPØRT  
 21 MANUFACTUR CATALØGU ABSTRACT \* INDEX \* PATENT  
 212 \* \* CATEGØRIZ \* PERCENT \* USE \* FUNCT GRØUP \*  
 213 ENGINE CHEMIC CØLSULT DESIGN \* DEVELOP FACILIT  
 214 PLAN \* INFØRMA RETRIEV

Figure IV-10(b)

Results of STEM

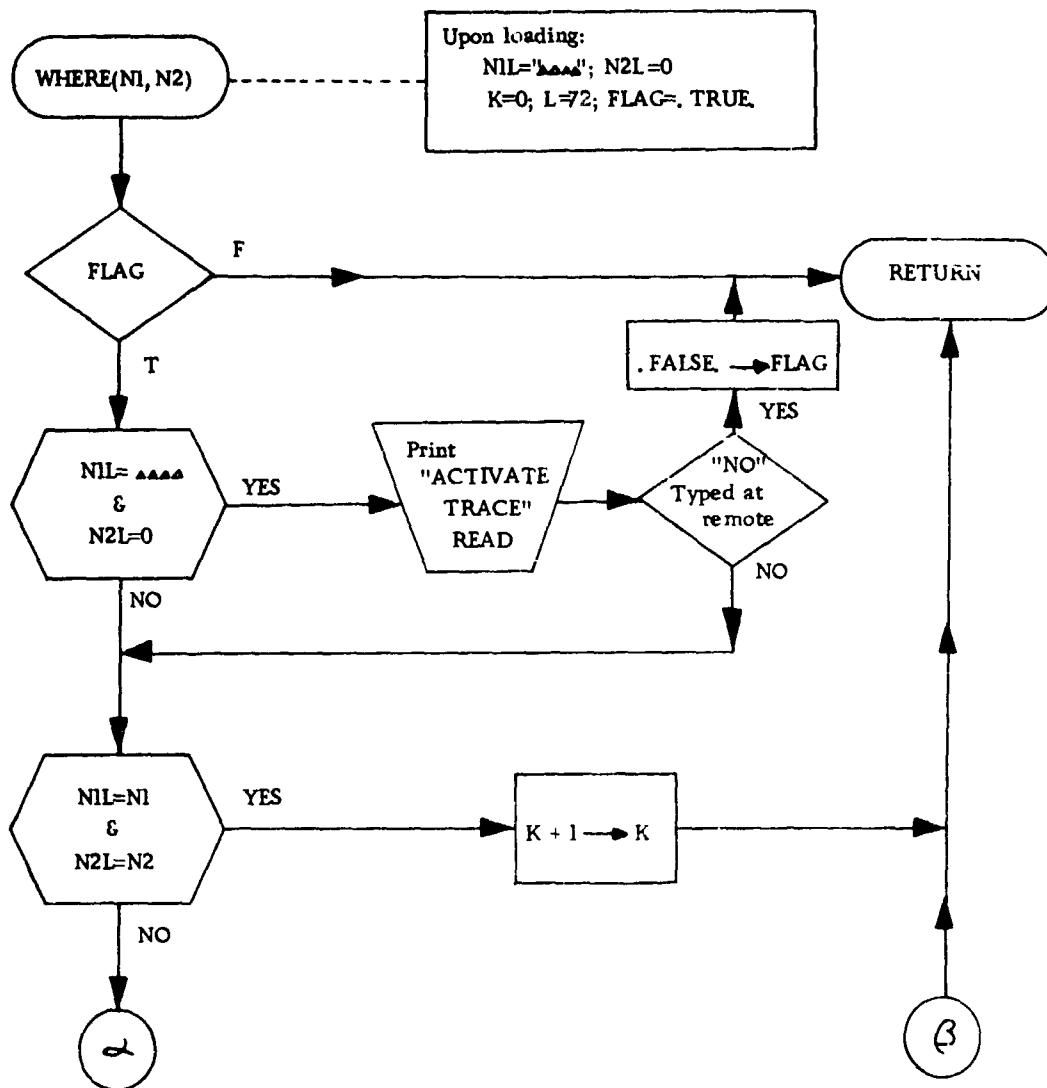


Figure IV-11 (cont'd)  
WHERE

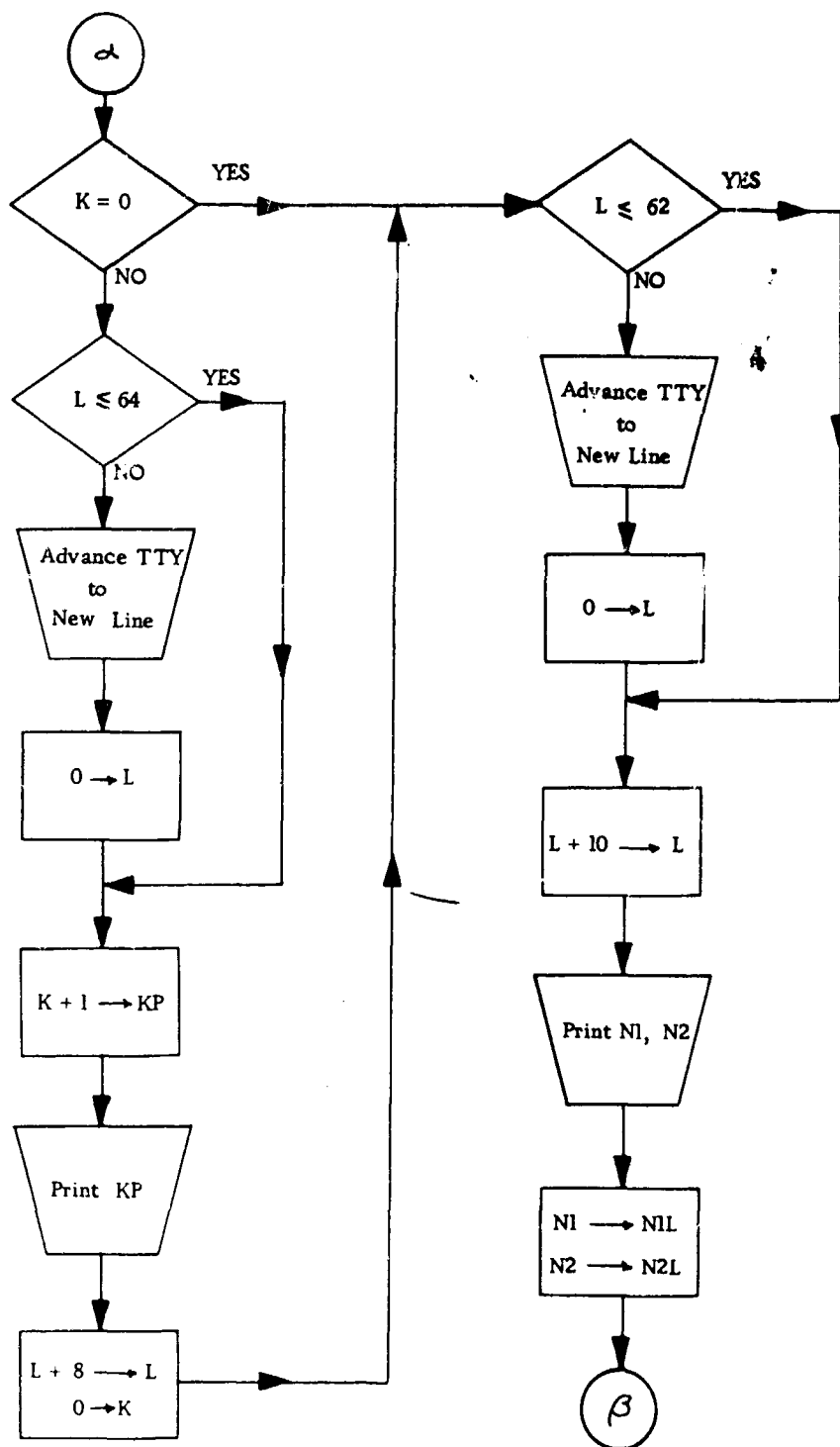


Figure IV-11 (concluded)

WHERE

2. In a loop resulting in repeated calls to WHERE with unchanging arguments, repeated printing of the arguments would be unnecessary, wasteful and annoying. Therefore the arguments are printed once upon entry to the loop, and upon exit those arguments are followed by: "\*kk/", where kk is the number of times the loop was executed.

#### IV. 9. 2     Method

The flowchart (Figure IV-11) and listing (Figure VI-9) explain this straightforward subroutine.

#### IV. 9. 3     Deactivation

In the code delivered to RADC, the subroutine has been altered so that the query "Activate Trace?" is not printed and no trace is supplied. This is done by adding two lines to WHERE:

```
99035      FLAG=.FALSE.; PRINT 500
99036 500  FORMAT (2H & (E))
```

#### IV.10        FUNCTION YESNO(I)

Many system-generated queries must be answered either "yes" or "no". This subroutine reads a string from the remote terminal and sets its arguments to one if "yes" was read or zero if "no" was read. The sophisticated user is allowed the word "options", which sets the argument to minus one; any other response causes the system to ask the user to 'ANSWER "YES" OR "NO".', and repeats the query.

In many applications it is useful to call such a routine as an arithmetic function, so that the statement

```
IF (YESNO(I) ) 1, 2, 3
```

branches to 1 for "options", to 2 for "no" and to 3 for "yes".

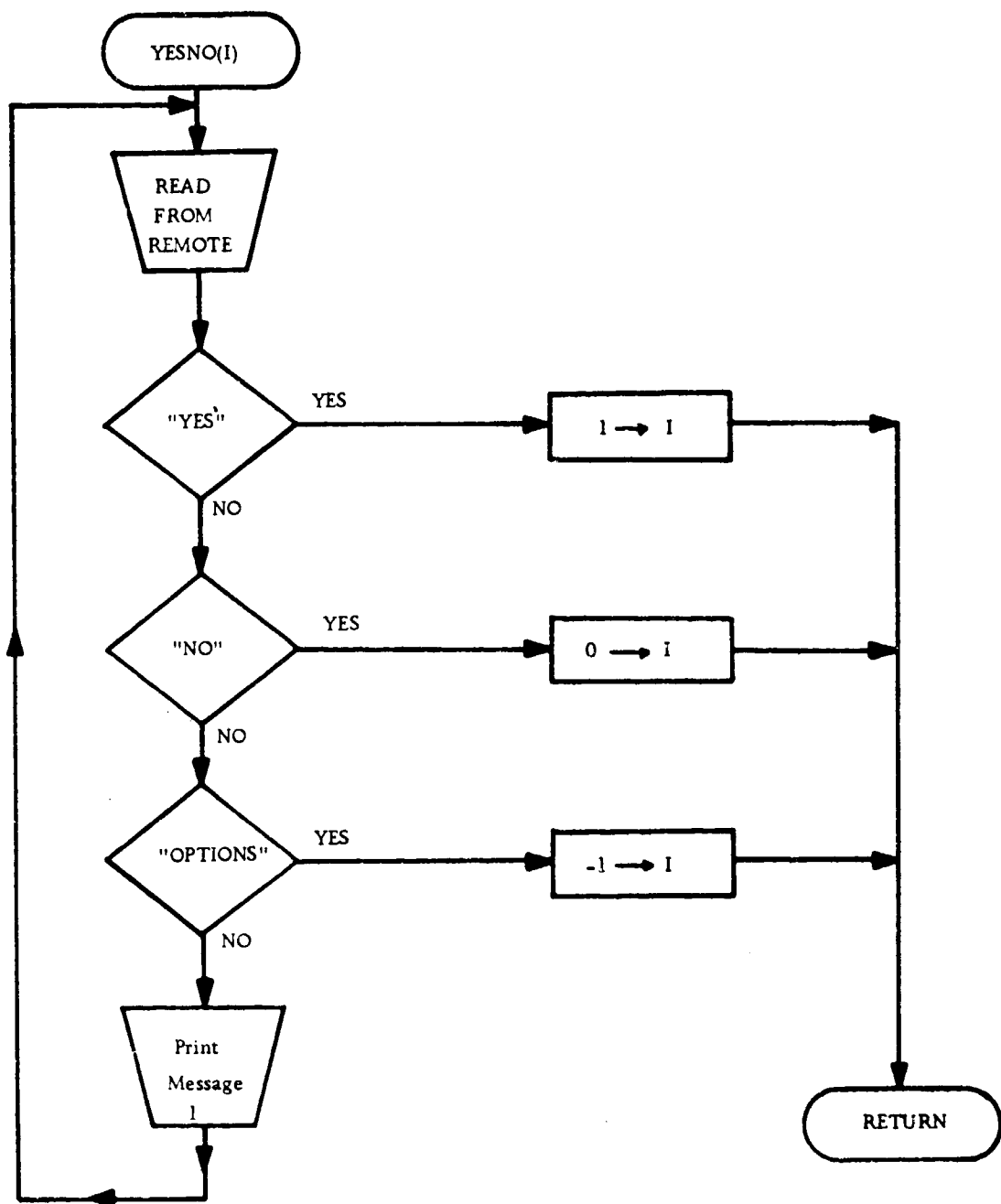


Figure IV-12  
YESNO

## SECTION V

### FURTHER WORK

The work performed on this project has resulted in the design and partial implementation of an on-line system that promises to add new important capabilities to on-line information retrieval systems. For this promise to be fulfilled, the investigation and implementation effort should be continued.

Of first importance is the implementation of the complete On-Line System itself. Once the System is implemented, it will provide the best possible tool for experimentation with the techniques of automatic indexing within an on-line environment. Estimates can be made of such factors as expected response time, precision, and relevance, and valuable experimentation can be conducted using batch programs. However, the combined impact upon the user of all these factors can be determined only by constructing the entire System. And it is this total impact on the user that determines the utility of these techniques.

Simultaneously with the implementation, experiments should be performed to determine optimum settings for the various features of the indexing programs that can be varied parametrically. The value of such experimentation is vividly illustrated by the discussion of the dictionary of 900 stems (Figure III-4) in subsection III. 2. 3. By studying the effect of variation of these various factors upon the results obtained, the System can be "tuned" to maximize its performance.

## SECTION VI

### LISTINGS

This Section contains listings of all programs and subprograms that comprise the dialogue processor and its associated supporting software. The Directory of Programs and Subprograms in Figure IV-1 serves to index this Section.

```

70000C      DOCK 11/10/69
70010C      FUNCTION DOCK GETS TITLES, AUTHORS OR ABSTRACTS ONE LINE AT
70020C      A TIME, -RETURNING FALSE WHEN ALL LINES RETURNED.
70030      FUNCTION DOCK(CODE,I,ARRAY,COUNT)
70040      LOGICAL DOCK
70040      COMMON TERSE, NNN(100), TEMP1(50), TEMP2(50), TEMP3(50),
70060      TEMP4(50), TEMP5(50), TEMP6(50), QUERY1(50), QUERY2(50)
70070      , PRE1(5,25), PRE2(3,25), PRE3(3,25), PRE4(3,25)
70080      , LGSTACN0
70090      INTEGER TEMP1, TEMP2, TEMP4, QUERY1, PRE3
70100      ASCII PRE1,PRE2
70110      LOGICAL TERSE, TEMPS, TEMP6
70120      ASCII ARRAY(18),TEMP0(18)
70130      LOGICAL FLAG
70140      INTEGER CODE,COUNT
70150      DATA FLAG/.FALSE./
70160      100 IF(LGSTACN0.GE.1 .AND. CODE.GE.1 .AND. CODE.LE.3) GO TO 400
70170      200 PRINT 201, CODE, I
70180      201 FORMAT("-ERROR IN DOCK AT CODE, I:", 2I20)
70190      300 DOCK=.FALSE.; FLAG=.FALSE.; COUNT=0; RETURN
70200C      TEST IF ON OLD SEQUENCE--FIND DESIRED DOCUMENT IF NOT.
70210      400 IF(FLAG) GO TO 700
70215      READ("DATA1",501) ARRAY
70220      BEGIN FILE "DATA1"
70230      J=0
70240      500 READ("DATA1",501,END=200) (ARRAY(K),K=1,18)
70250      501 FORMAT(18A4)
70260      IF(ARRAY(1).NE."**11") GO TO 500
70270      J=J+1
70280      IF(J.NE.1) GO TO 500
70290C      CORRECT DOCUMENT LOCATED. FIND DESIRED INFORMATION. CODE=1,2
70300C      3 AS TITLE, AUTHOR OR ABSTRACT.
70310      600 READ("DATA1",501,END=200) (ARRAY(K),K=1,18)
70320      IF(CODE.EQ.1) GO TO 800

```

Figure VI-1 Function DOCK (cont'd)

```

70330      IF(.NOT.((CODE.EQ.2 .AND. ARRAY(1).EQ."**AU") .OR.
70340      (CODE.EQ.3 .AND. ARRAY(1).EQ."**AB")) ) GO TO 600
70350      READ("DATA1",501,END=200) (ARRAY(K),K=1,18)
70360      GO TO 800
70370      DO 701 K=1,18
70380      701 ARRAY(K)=TEMP0(K)
70390      800 READ("DATA1",501,END=200) (TEMP0(K), K=1,18)
70400      IF(TEMP0(1).GE."**AA".AND.TEMP0(1).LE."**ZZ") GO TO 900
70410      DOCK=.TRUE.; FLAG=.TRUE.; GO TO 950
70420      DOCK=.FALSE.; FLAG=.FALSE.
70430      DO 960 J=1,18
70440      960 IF(ARRAY(J).NE." ") COUNT=J
70450      RETURN; END

```

Figure VI-1 Function DOCK (concluded)

82000C	FUNCTION 'LENGTH' FOR STEM ANALYSIS ROUTINE.
82010	FUNCTION LENGTH(X,Y,Z,I)
82020	INTEGER A,B(4),X(5),Y(5),Z(5),PL0P
82030	DATA L,L2/0000040040040.0040040040040/
82040	DATA L3/0040000000000/
82050 1	D0 2 J=1,5
82060	Z(J)=L2
82070 2	Y(J)=L2
82080	J=5
82090 3	IF(X(J).NE.L2) G0 T0 4
82100	J=J-1
82110C	TEST T0 SEE IF INPUT STRING NULL (BLANKS).
82120	IF (J.GT.0) G0 T0 3
82130	LENGTH=0
82140	RETURN
82150C	HERE WHEN J=N0. 0F WORDS CONTAINING AT LEAST ONE CHARACTER.
82160C	ALSO IS INDEX 0F LAST SUCH WORD.
82170 4	LTH=4*(J-1)
82180	A=X(J)
82190	ASSIGN 7 T0 IB
82200 5	K=134217728
82220	M=0
82230 6	IF(A.LT.K) N=0
82235	IF(A.GE.K) N=A/K
82240	A=A-N*K
82242	M=M+1
82244	B(M)=N*134217728+L
82246	K=K/512
82248	IF(M.LT.4) G0 T0 6
82250C	B HAS ONE CHARACTER PER ELEMENT
82260	G0 T0 IB, (7,13)
82270 7	D0 8 M=1,4
82280	IF(B(M).EQ.L2) G0 T0 9

Figure VI-2 Function LENGTH (cont'd)

```

82290 8 LTH=LTH+1
82300 9 LENGTH=LTH
82310C STRING LENGTH COMPUTED. NOW SPLIT UP.
82320 PLØP=LTH-1
82330C IF I>STRING LENGTH, PLØP<0 PUTS STRING IN Y.
82340 JX=1; JY=1; JZ=1; JCXT=0
82350 10 IF(PLØP.LT.4) GØ TØ 11
82360 Z(JZ)=X(JX)
82370 PLØP=PLØP-4
82380 JCXT=JCXT+4
82390 JX=JX+1; JZ=JZ+1; GØ TØ 10
82400 11 JCXW=0; JCYW=0; JCZW=0
82410 12 ASSIGN 13 TØ 1B; A=X(MINO(JX,5)); GØ TØ 5
82420 13 NRCX=0
82430 14 IF(JCXT.GE.LTH) RETURN
82440 JCXT=JCXT+1; NRCX=NRCX+1
82450 IF(PLØP.GT.0) GØ TØ 15
82460 K=1
82461 IKYPØØ=JCYW
82462 998 IF(ΙΚYPØØ.EQ.0) GØ TØ 999
82463 K=K*512; ΙΚYPØØ=ΙΚYPØØ-1; GØ TØ 998
82464 999 CONTINUE
82470 Y(JY)=(Y(JY)-L3/K)+(B(NRCX)-L)/K
82480 JCYW=JCYW+1
82490 IF(JCYW.LT.4) GØ TØ 16
82500 JY=JY+1
82510 JCYW=0
82520 GØ TØ 16
82530C BLOCK STARTING WITH 10 MOVES GROUPS ØF 4 CHARACTERS INTO Z.
82540C PART ABOVE MOVES CHAR. TØ Y. BELOW CHAR. TØ Z UNBLOCKED.
82550 15 K=1
82551 ΙΚYPØØ=JCZW
82552 887 IF(ΙΚYPØØ.EQ.0) GØ TØ 888

```

Figure VI-2 Function LENGTH (cont'd)

```

82553      K=K*512; IKYP00=IKYP00-1; G0 T0 887
82554      888 CONTINUE
82560      Z(JZ)=(Z(JZ)-L3/K)+(B(NRCX)-L)/K
82570      JCZW=JCZW+1; PL0P=PL0P-1
82580      IF(JCZW.LT.4) G0 T0 16
82590      JZ=JZ+1
82600      JCZW=0
82610      16 IF(NRCX.LT.4) G0 T0 14
82620      JX=JX+1
82630      IF(JX.GT.5) RETURN
82640      G0 T0 12
82650      END

```

Figure VI-2 Function LENGTH (concluded)

```

75010C FUNCTION TAKES STEM IN PRE2 AND PLACES CON. VECT. IN PRE.
75015C VALUE TRUE IF STEM IN FILE 'DICTNRY'.
75020 FUNCTION LOOKUP(I)
75030 COMMON TERSE, N0N0(100), TEMP1(50), TEMP2(50), TEMP3(50),
75040& TEMP4(50), TEMP5(50), TEMP6(50), QUERY1(50), QUERY2(50)
75050& , PRE1(5,25), PRE2(3,25), PRE3(3,25), PRE4(3,25)
75060& , LGSTACN0
75070 INTEGER TEMP1, TEMP2, TEMP4, QUERY1, PRE3
75080 ASCII PRE1,PRE2
75090 LOGICAL TERSE, TEMP5, TEMP6
75100C DICTIONARY LOOKUP FUNCTION PUTS CONCEPT VECTOR IN PRE-
75110C TABLE CORRESPONDING TO STEM IN I-TH LOCATION OF TABLE.
75120C RETURNS WITH FALSE VALUE IF STEM NOT IN DICTIONARY.
75130 LOGICAL TOP, BOTTOM, LOOKUP
75140 DIMENSION WT(3,500)
75150 ASCII TABLE(3,500)
75160 INTEGER CON(3,500)
75170 DATA L,L0LD,MAX/0,-100,500/
75180C SEE IF STEMS ARE IN CORE. LOAD IF NOT.
75190 100 IF( L.NE.0) GO TO 300
75200 200 IF(L.NE.0) BEGIN FILE "DICTNRY"
75210 BOTTOM=.TRUE.; GO TO 220
75220C IF MORE STEMS NEEDED, OVERLAY THEM.
75230 210 BOTTOM=.FALSE.
75240 220 TOP=.FALSE.
75250C CLEAR CORE DICTIONARY & VECTORS BEFORE LOADING.
75260 DO 221 N1=1,MAX
75270 DO 221 N2=1,3
75280 TABLE(N2,N1)=" "
75290 CON(N2,N1)=0
75300 221 WT(N2,N1)=0.0
75310C READ UNTIL TABLES FULL OR FILE EXHAUSTED.
75320 L=1
75330 230 READ("DICTNRY", 231, END=240) (TABLE(J,L),J=1,3), (CON(J,L),
75340& WT(J,L), J=1,3)

```

Figure VI-3 Function LOOKUP (cont'd)

```

75350 231 FORMAT(3A4, 1X, 3(I4,F9.5))
75360 IF(L.GE.MAX) GO TO 300
75370 L=L+1; GO TO 230
75380C ALL FILE READ.
75390 240 L=L-1; TOP=.TRUE.; GO TO 300
75400C DOES QUERY STEM COLLATE BEFORE DICTIONARY SEGMENT IN CORE?
75410 300 IF (PRE2(1,1) .LT. TABLE(1,1) ) GO TO 305
75420 IF (PRE2(1,1) .GT. TABLE(1,1) ) GO TO 320
75430 IF (PRE2(2,1) .LT. TABLE(2,1) ) GO TO 305
75440 IF (PRE2(2,1) .GT. TABLE(2,1) ) GO TO 320
75450 IF (PRE2(3,1) .GE. TABLE(3,1) ) GO TO 320
75460C STEM COMES BEFORE PART OF DICTIONARY IN CORE. IF THE FIRST
75470C PART IS IN CORE, THEN STEM NOT IN DICTIONARY.
75480 305 IF(.NOT.BOTTOM) GO TO 200
75490C HERE ON ALL UNSUCCESSFUL LOOKUPS.
75500 310 LOOKUP=.FALSE.
75510 DO 311-N1=1,3
75520 PRE3(N1,1)=0
75530 311 PRE4(N1,1)=0.0
75540 RETURN
75550C PERFORM SIMILIAR FUNCTION TO SEE IF STEM FOLLOWS HIGHEST
75560C DICTIONARY SEGMENT IN CORE & IF THAT IS LAST SEGMENT.
75570 320 IF (PRE2(1,1) .GT. TABLE(1,L) ) GO TO 325
75580 IF (PRE2(1,1) .LT. TABLE(1,L) ) GO TO 400
75590 IF (PRE2(2,1) .GT. TABLE(2,L) ) GO TO 325
75600 IF (PRE2(2,1) .LT. TABLE(2,L) ) GO TO 400
75610 IF (PRE2(3,1) .LE. TABLE(3,L) ) GO TO 400
75620 325 IF(TOP) GO TO 310
75630 GO TO 210
75640C K IS THE SMALLEST POWER OF 2 THAT IS > CURRENT IN-CORE
75650C SEGMENT LENGTH. COMPUTE ONLY IF IT HAS CHANGED.
75660 400 IF(L.EQ.LOLD) GO TO 430
75670 410 K=1; LOLD=L
75680 420 IF(K.GT.L) GO TO 430

```

Figure VI-3 Function LOOKUP (cont'd)

```

75690      K=K+K; G0 T0 420
75700C     NOW START BINARY SEARCH. IK IS INDEX AND J IS DELTA.
75710 430 IK=K/2
75720 440 J=IK/2
75730 445 IF(IK.GT.L) G0 T0 490
75740C     ARE WE LOOKING TOO HIGH IN THE TABLE?
75750      IF(PRE2(1,I).LT.TABLE(1,IK)) G0 T0 490
75760      IF(PRE2(1,I).EQ.TABLE(1,IK).AND.PRE2(2,I).LT.TABLE(2,IK))
75765&      G0 T0 490
75770      IF((PRE2(1,I).EQ.TABLE(1,IK).AND.PRE2(2,I).EQ.TABLE(2,IK))
75775&      .AND.PRE2(3,I).LT.TABLE(3,IK)) G0 T0 490
75780C     EITHER A MATCH IS FOUND OR WE'RE LOOKING TOO LOW.
75790 460 D0 461 N1=1,3
75800      IF (PRE2(N1,I) .NE. TABLE(N1,IK) ) G0 T0 480
75810 461 CONTINUE
75820C     HERE IF SUCCESSFUL. PUT CONCEPT VECTOR IN PRE & RETURN.
75830 470 D0 471 N1=1,3
75840      PRE3(N1,I)=CON(N1,IK)
75850 471 PRE4(N1,I)=WT(N1,IK)
75860      LOOKUP=.TRUE.; RETURN
75870 480 IK=IK+J; G0 T0 500
75880 490 IK=IK-J
75890 500 IF(J.EQ. 0) G0 T0 310
75900      J=J/2; G0 T0 445; END

```

Figure VI-3 Function LOOKUP (concluded)

```

78000C SUBROUTINE 'NUMBER' IDENTIFIES DOCUMENTS, REQUIRES COMMON
78010C MESSAGE FILE PLUS YESN0, 0UT, PLUCK, PUT, LENGTH
78020C AND ZORCH.
78030 SUBROUTINE NUMBEK(F1,AKG,NR,NP)
78040C SHARED COMMON STORAGE FOR THE DIALOGUE PROCESSOR.
78050 COMMON TERSE, N0N0(100), TEMP1(50), TEMP2(50), TEMP3(50),
78060& TEMP4(50), TEMP5(50), TEMP6(50), QUERY1(50), QUERY2(50),
78070& PRE1(5,25), PRE2(3,25), PRE3(3,25), PRE4(3,25)
78080& , LGSTACN0
78090 INTEGER TEMP1, TEMP2, TEMP4, QUERY1, PRE3
78100 ASCII PRE1, PRE2
78110 LOGICAL TERSE, TEMP5, TEMP6
78120 INTEGER F1,ARG,RNG,R,P,ACN0,ZORCH,YESN0,HI,N(2)
78130 ASCII X(18), Y(5), Z(5)
78140 DATA RNG/0/
78150 IF(F1.GT.0) G0 T0 130
78160 IF(F1.LT.0) G0 T0 90
78170C ENTRY IS NOT ONE OF A SEQUENCE.
78180 10 CALL 0UT(2)
78190 20 CALL PLUCK(" " ,X,L,1,NR,NP)
78200C READ TTY INPUT.
78210 IF(L.GT.6) G0 T0 50
78220C L>6 IMPLYS ERROR.
78230 IF(X(1).EQ."ALL") G0 T0 180
78240 P=0;R=1;N(1)=0;N(2)=0;J=1;K=0
78250C SCAN INPUT FROM RIGHT.
78260 21 ND=LENGTH(X,Y,Z,1)
78270 IF(Y(1).EQ."A") G0 T0 59
78280C INPUT IS RANGE, SMALLER IN N(1).
78290 IF(Y(1).NE."-") G0 T0 22
78300 P=-1;R=2;K=0; G0 T0 23
78310 22 IF(ZORCH(Y(1)).GT.9) G0 T0 50
78320 N(R)=N(R)+(MAX0(1,10**K))*(ZORCH(Y(1)))
78330 K=K+1
78340 23 J=J+1

```

Figure VI-4 Subroutine NUMBER  
Function ZORCH (cont'd)

```

78350 IF(J.GT.L) G0 T0 30
78360 D0 24 M=1,5
78370 24 X(M)=Z(M)
78380C RETURN FOR NEXT CHARACTER.
78390 G0 T0 21
78400 30 IF(P.LT.O) G0 T0 120
78410C T0 120 IF RANGE. SEARCH FOR SINGLE TEMP. ID. IF NOT.
78420 D0 31 M=1,50
78430 IF(TEMP2(M).EQ.N(1)) G0 T0 40
78440 31 CONTINUE
78450 ACN0=-1; G0 T0 60
78460 40 ACN0=TEMP1(M); G0 T0 60
78470 50 CALL OUT(5); G0 T0 10
78480 59 ACN0=N(1)
78490 60 IF( ACN0.GT.O .AND. ACN0.LE.LGSTACN0 ) G0 T0 80
78500 70 CALL OUT(4); G0 T0 90
78510 80 F1=-1; ARG=ACN0; RETURN
78520 90 CALL OUT(3)
78530 IF(YESN0(P).GT.O) G0 T0 10
78540C NO MORE WANTED
78550 100 F1=0
78560 110 RETURN
78570 120 I=N(2); HI=N(1); F1=0; KNG=1; G0 T0 140
78580C RANGE OF TEMP. NOS. OR "ALL".
78590 130 IF(RNG.NE.1) G0 T0 190
78600 140 D0 141 M=1,50
78610 IF(TEMP2(M).EQ.1) G0 T0 150
78620 141 CONTINUE
78630 IF(I.GE.HI .AND. F1.EQ.O) G0 T0 70
78640 IF(I.LT.HI) G0 T0 142
78650 F1=-1; G0 T0 90
78660 142 I=I+1; G0 T0 140
78670 150 ARG=TEMP1(M); I=I+1
78680 IF(I.GT.HI) G0 T0 160
78690 F1=F1+1; G0 T0 110

```

Figure VI-4 Subroutine NUMBER  
Function ZORCH (cont'd)

```

78700 160 F1=-1; G0 T0 110
78710 180 RNG=0; I=1; F1=0
78720C ALL TEMP FILE WANTED.
78730 190 IF (I.GT.50) G0 T0 220
78740 IF(TEMP1(I).NE.0) G0 T0 210
78750 I=I+1; G0 T0 190
78760 210 ARG=TEMP1(I); I=I+1; F1=F1+1; G0 T0 110
78770 220 IF(F1.NE.0) G0 T0 230
78780C F1=0 & I>50 IMPLYS TEMP FILE EMPTY.
78790 CALL OUT(6); G0 T0 70
78800 230 F1=0; G0 T0 110
78810 END
78820C
78830C
78840C
78850C
78860
78870 INTEGER ZORCH
78880 DATA LEFT,LEFTL/0060040040040,0071040040040/
78890 IF(K.LT.LEFT .OR. K.GT.LEFTL) G0 T0 1
78900 ZORCH=(K-LEFT)/134217728; RETURN
78910 1 ZORCH=100; RETURN; END

```

FUNCTION 'ZORCH' IS USED BECAUSE OF RESTRICTIONS IN SOME  
VERSIONS OF THE LANGUAGE--IT AVOIDS MIXING MODES.

Figure VI-4 Subroutine NUMBER  
Function ZORCH (concluded)

```

80000C      OUT 11/10/69
80010C      SUBROUTINE OUT PRINTS FROM FILE 'MESSAGES'. TERSE FORMS FIRST.
80020      SUBROUTINE OUT(J)
80030      COMMON TERSE, N0N0(100), TEMP1(50), TEMP2(50), TEMP3(50),
80040      TEMP4(50), TEMP5(50), TEMP6(50), QUERY1(50), QUERY2(50)
80050      , PRE1(5,25), PRE2(3,25), PRE3(3,25), PRE4(3,25)
80060      , LGSTACN0
80070      INTEGER TEMP1, TEMP2, TEMP4, QUERY1, PRE3
80080      ASCII PRE1,PRE2
80090      LOGICAL TERSE, TEMP5, TEMP6
80100      INTEGER I,TT
80110      ASCII A(67)
80120      DATA A(67)/"/"/
80130      1 READ('MESSAGES',2,END=100) M,T,L,(A(K),K=1,66)
80140      2 FORMAT(I2,I1,I2,I1,66A1)
80150      IF(M.NE.J) GO TO 1
80160      IF((T.EQ.0) .AND. (.NOT. TERSE)) GO TO 1
80170      TT=T
80180      3 I=0
80190      4 IF(A(I+1).EQ."/".OR.I.LE.66) GO TO 5
80200      I=I+1 GO TO 4
80210      5 PRINT 6,(A(K),K=1,I)
80220      6 FORMAT(1H,66A1)
80230      READ('MESSAGES',2,END=200)M,T,L,(A(K),K=1,66)
80240      IF(M.EQ.J.AND.T.EQ.TT) GO TO 3
80250      BEGIN FILE 'MESSAGES'; RETURN
80260      100 PRINT 300,J; J=-1; GO TO 200
80270      300 FORMAT("-ERR0R IN 'OUT' SUBROUTINE AT MESSAGE ", I4)
80280      END

```

Figure VI-5 Subroutine OUT

```

83000C SUBROUTINE 'PLUCK' GETS STRINGS FROM FILES. IT REQUIRES FUNCT-
83010C TION 'PUT', WHICH PACKS ALPHA INFORMATION.
83020 SUBROUTINE PLUCK(FILE,ARRAY,L,I,K,P)
83030 INTEGER K,OLDR,P,PUT,START
83040 FILENAME FILE
83045 FILENAME OLDFILE
83047 ASCII FILE1,FILE2
83048C BECAUSE CURRENT KADC VERSION HAS NOT BEEN CHANGED TO ALLOW
83049C FILENAME COMPARISONS, ONLY FIRST 4 CHARACTERS USED.
83050 ASCII ARRAY(18), DELIM(11), BLANKS, BUF(72)
83060 DATA BLANKS/" "/
83070 DATA OLDFILE/"",/
83080 DATA DELIM/0040040040,0054040040040,0056040040040,
83090 0042040040040,0041040040040,0073040040040,0072040040040,
83100 0050040040040,0051040040040,0077040040040,0055040040040/
83110C LIMIT=1,10 OR 11: CONTROLS SELECTION OF DELIMIT CHARACTERS.
83120 IMAG=IABS(I)
83130 LIMIT=1+9*(IMAG/2)+IMAG/3
83140C CHECK FOR ILLEGAL PARAMETERS
83150 IF(.NOT.(K.LT.1.OR.IMAG.GT.3.OR.P.LT.1.OR.P.GT.72)) GO TO 10
83160 PRINT 1,R,P,I
83170 1 FORMAT("PAR. ERROR IN 'PLUCK': K,P,I", 3I12)
83180 L=0;P=-100;RETURN
83190C INITIALIZE, THEN SET UP FOR CONTINUATION ON FILE SWITCH IF I=0
83200 10 DO 11 J=1,18
83210 11 ARRAY(J)=BLANKS
83220 IF(I.NE.0) GO TO 20
83230 OLDFILE=FILE
83240 OLDR=R
83250 BACKSPACE FILE
83260 READ(FILE,12,END=60) BUF
83270 12 FORMAT(72A1)
83280 RETURN
83290C RESET IF NEW FILE NOT PRECEDED BY I=0, THEN GET RECORD.
83300 20 FILE1=FILE; FILE2=OLDFILE

```

Figure VI-6 Subroutine PLUCK (cont'd)

```

83305 IF(FILE1.EQ.FILE2) GO TO 30
83310 OLDFILE=FILE
83320 GO TO 31
83330 30 IF(R.EQ.OLDR) GO TO 70
83340 31 IF(FILE1.EQ.BLANKS) GO TO 40
83341 READ(FILE,12,END=60) BUF
83342 BEGIN FILE FILE
83350 OLDR=0
83350 40 IF(R.EQ.OLDR) GO TO 70
83370 50 READ(FILE,12,END=60) BUF
83380 OLDR=OLDR+1; GO TO 40
83390C RETURN WITH L=R=P=0 IF END REACHED.
83400 60 L=0;R=0;P=0;RETURN
83410C NOW STEP THROUGH BUFFER TO FIND NEXT DELIMITED STRING.
83420 70 L=0;START=P
83430 80 IF(P.GT.72) GO TO 100
83440 DO 81 J=1,LIMIT
83450 IF(BUF(P).EQ.DELIM(J)) GO TO 110
83460 81 CONTINUE
83470C HERE IF NON-DELIM FOUND
83480 L=PUT(BUF(P),L,ARRAY)
83490 90 P=P+1; GO TO 80
83500C HERE IF END OF RECORD; AT 110 IF OTHER DELIM FOUND.
83510 100 P=1
83520 R=R+1
83530 110 IF(L.NE.0) GO TO 120
83540 IF(R.EQ.OLDR) GO TO 90
83550 GO TO 50
83560C GOOD STRING FOUND. IGNORE IF 1<0 & FIRST STRING OF RECORD.
83570 120 IF(1.LT.0.AND.START.EQ.1) GO TO 10
83580 RETURN;END

```

Figure VI-6 Subroutine PLUCK (concluded)

```

838000 FUNCTION 'PUT' PUTS CHARACTER A INTO L+1 CHAR. POSITION
838100 OF ARRAY. ARRAY MUST BE INITIALLY CLEARED TO BLANKS. VALUE
838200 RETURNED IS L+1; L AND A ARE UNCHANGED.
838300 FUNCTION PUT(A,L,ARRAY)
838400 INTEGER PUT
838500 INTEGER A,ARRAY(13),B1,B2,PUT
838600 DATA B1,B2/0000040040040,0040000000000000/
838700 N=L+1
838800 PUT=N
838900 JC=M0D(L,4)
83900 JW=L/4+1
83910 K=1
83911 998 IF(JC.EQ.0) G0 T0 999
83912 K=K*512; JC=JC-1; G0 T0 998
83913 999 IF(B2.GE.K) ARRAY(JW)=ARRAY(JW)-B2/K
83914 IF((A-B1).GE.K) ARRAY(JW)=ARRAY(JW)+(A-B1)/K
83920 RETURN; END

```

Figure VI-7 Function PUT

```

79000C SUBROUTINE 'STEM' FOR STEM ANALYSIS. W IS WORD INPUT, S IS STEM
79010C OUTPUT, COUNT IS LENGTH OF STEM OR ZERO IF FUNCTION WORD.
79020 SUBROUTINE STEM(W,S,COUNT)
79030 ASCII A(3), S(5), W(5), X(5), Y(5), V(5), CCM(250,3),
79031 STM(4,50), BLANKS
79040 LOGICAL FLAG
79050 INTEGER COUNT,CODE,PASS,F(5)
79060 DATA F,IC/6*1/
79070 DATA CCM,STM,BLANKS,V,FLAG/951* " ,
79080 "AA","II","EE","00","UU",.FALSE./
79090C CHECK IF COMMON WORDS AND STEMS LOADED--LOAD THEM IF NOT.
79100 1 IF(FLAG) GO TO 10
79110 FLAG=.TRUE.
79120 2 READ('WORDS',3,END=9) CODE,A
79130 3 FORMAT(11,4X,3A4)
79140 IF(CODE.NE.0) GO TO 5
79150 DO 4 J=1,3
79160 4 CCM(IC,J)=A(J)
79170 IC=IC+1; GO TO 6
79180 5 IF(CODE.LT.1.OR.CODE.GT.4) GO TO 7
79190 JS=F(CODE); F(CODE)=F(CODE)+1
79200 STM(CODE,JS)=A(1)
79210 6 IF(IC.LE.250.AND.JS.LT.50) GO TO 2
79220 7 PRINT 8
79230 8 FORMAT("--ERROR IN FILE 'WORDS'")
79240 9 CLOSE FILE "WORDS"
79250C MAKE IC THE NUMBER OF COMMON WORDS STORED
79260 IC=IC-1
79270C NOW CHECK TO SEE IF INPUT IS COMMON WORD (UP TO 12 CHARACTERS)
79280 10 DO 12 I=1,IC
79290 DO 11 J=1,3
79300 IF(W(J).NE.CCM(I,J)) GO TO 12
79310 11 CONTINUE
79320 GO TO 13
79330 12 CONTINUE

```

Figure VI-8 Subroutine STEM (cont'd)

```

79340      GO TO 15
79350C     HERE IF INPUT IS COMMON WORD.
79360 13   COUNT=0
79370      DO 14 J=1,5
79380 14   S(J)=BLANKS
79390      RETURN
79400C     NOW DO STEM ANALYSIS. FIRST INITIALIZE--X IS STRING PROC.
79410 15   PASS=0
79420      DO 16 J=1,5
79430 16   X(J)=W(J)
79440C     START OF FULL PASS
79450 17   I=4
79460C     CHECK FOR SUFFIX OF LENGTH 1.
79470 18   L=LENGTH(X,Y,S,I)
79480      IF(Y(I).EQ.BLANKS) GO TO 20
79490      DO 19 J=1,50
79500      IF(STM(I,J).EQ.BLANKS) GO TO 20
79510      IF(STM(I,J).EQ.Y(I)) GO TO 22
79520 19   CONTINUE
79530 20   IF(I.EQ.1) GO TO 25
79540 21   I=I-1; IF(I.LE.0) GO TO 28
79550      GO TO 18
79560C     HERE IF STEM FOUND. REMOVE IT UNLESS WORD TOO SHORT.
79570 22   IF(L-I.LT.5) GO TO 21
79580      DO 23 J=1,5
79590 23   X(J)=S(J)
79600C     IF WORD ENDED IN 'LY' CHECK FOR 'I'.
79610      IF(Y(I).NE."LY") GO TO 25
79620      L=LENGTH(X,Y,S,I)
79630      IF(Y(I).NE."I") GO TO 25
79640      DO 24 J=1,5
79650 24   X(J)=S(J)
79660C     CHECK FOR DOUBLE CONSONANT.
79670 25   L=LENGTH(X,F,S,2)

```

Figure VI-8 Subroutine STEM (cont'd)

```

79680      IF(F(1)/262144.NE.(F(1)/134217728)*513) GO TO 28
79690      L=LENGTH(X,Y,S,2)
79700C     V CONTAINS 'AA' ... 'UU'.
79710      DO 26 J=1,S
79720      IF(Y(1).EQ.V(J)) GO TO 28
79730      26 CONTINUE
79740      L=LENGTH(X,Y,S,1)
79750      DO 27 J=1,S
79760      27 X(J)=S(J)
79770C     A PASS HAS BEEN MADE--TWO ALLOWED. USE 'LENGTH' TO PUT STEM
79780C     IN S AND COUNT ITS LENGTH IF TWO PASSES MADE.
79790      28 IF(PASS.NE.1) GO TO 29
79800      COUNT=LENGTH(X,Y,S,0)
79810      RETURN
79820      29 PASS=1, GO TO 17
79830      END

```

Figure VI-8 Subroutine STEM (concluded)

```

99000C      WHERE 11/10/69
99010      SUBROUTINE WHERE(N1,N2)
99020      LOGICAL FLAG, ASCII N1, NIL, A
99030      DATA NIL,N2L,K,L,FLAG/"",0,0,72,.TRUE./
99040      IF (FLAG) GO TO 2
99050      RETURN
99060      IF(NIL.NE."") ".0R.N2L.NE.0) GO TO 4
99070      PRINT 3
99080      FORMAT(" ACTIVATE TRACE?")
99090      READ:A
99100      IF (A.NE."N0") GO TO 4
99110      FLAG=.FALSE.)GO TO 1
99120      IF(N1.EQ.NIL.AND.N2.EQ.N2L) GO TO 11
99125      IF(K.EQ.0) GO TO 8
99130      IF(L.LE.64)GO TO 6
99140      PRINT 5
99150      FORMAT(1H )
99160      L=C
99170      PRINT 7,K
99180      FORMAT(2H*,I6,1H/)
99190      L=L+8
99200      K=0
99210      IF(L.LE.62) GO TO 9
99220      PRINT 5
99230      L=0
99240      L=L+10
99250      PRINT 10,N1,N2
99260      FORMAT(3H& ,A4,I4)
99270      NIL=N1
99280      N2L=N2
99285      GO TO 1
99287      K=K+1
99290      GO TO 1) END

```

Figure VI-9 Subroutine WHERE

81000C	FUNCTION YESN0(I) RETURNS VALUE IN NAME AND ARGUMENT.
81010	FUNCTION YESN0(I)
81020	INTEGER YESN0
81030	ASCII A
81040 1	READ(" ",2)A
81050 2	FORMAT(A4)
81060	IF (A.EQ."YES") G0 T0 3
81070	IF (A.EQ."N0") G0 T0 4
81080	IF (A.EQ."OPTI") G0 T0 5
81090	CALL OUT(I)
81100	G0 T0 1
81110 3	I=1;G0 T0 6
81120 4	I=0;G0 T0 6
81130 5	I=-1
81140 6	YESN0=I;RETURN;END

Figure VI-10 Function YESNO

```

999C *** ON-LINE DIALOGUE PROCESSOR 1/15/70
1000C DIALOGUE PROCESSOR--MAIN PROGRAM
1010 COMMON TERSE, N0N0(100), TEMP1(50), TEMP2(50), TEMP3(50),
1020 TEMP4(50), TEMP5(50), TEMP6(50), QUERY1(50), QUERY2(50),
1030 PRE1(5,25), PRE2(3,25), PRE3(3,25), PRE4(3,25), LGSTACN0
1040 LOGICAL TERSE, TEMPS, TEMP6, WFLG, SEEFLG, NEW0, RFLG, DEFLG,
1050 WRDFLG, D0CD0C, SKIP1, PRINT0, PRINT0, OPTION, FLAG, RAG,
1060 BAG, PS, LOOKUP, D0CK, FIRST
1070 INTEGER YESN0, F1, ARG, PUT, TEMP1, TEMP2, TEMP4, QUERY1, PRE3
1080 ASCII PRE1, PRE2, OPTIONS(13), VALUES(6), S(5), W(5),
1090 ARRAY(18)
1100 FILENAME TELETYPE
1110 DIMENSION WT(5), ICON(5)
1120 DATA OPTIONS/ "CHG", "CLR", "CON", "DDC", "DEL", "END", "D0C",
1130 "MOD", "OFF", "RET", "SEE", "WGT", "WRD"/
1140 DATA VALUES/ "1", "2", "3", "4", "5", "END"/-
1150 DATA T0

1160C SINCE INITIAL ENTRY OCCURS ONLY ONCE DURING USE, DATA STATE-
1170C MENTS INITIALIZE AND #100 COMES BELOW, WHERE COMMON INVOLVED.
1180C DATA JX,IX,NT,NT/0,3*1/
1190C DATA NEW0,RFLG,DEFLG,WFLG,SEEFLG,WRDFLG,D0CD0C/7*.FALSE./
1200C DATA SKIP1,PRINT0,PRINT0,OPTION/4*.FALSE./
1210C CALL OUT(74)
1220C INITIALIZE COMMON STORAGE
1230C ***** HIGHEST ACCESSION NUMBER SET HERE*****
1240C LGSTACN0=50; CALL WHERE("INIT",100)
1250C TERSE=.FALSE.
1260C D0 101 I=1,50
1270C N0N0(1)=0; N0N0(1+50)=0; TEMP3(1)=0.0
1280C TEMP1(1)=0; TEMP2(1)=0; TEMP4(1)=0; QUERY1(1)=0
1290C TEMPS(1)=.FALSE.; TEMP6(1)=.FALSE.; QUERY2(1)=0.0
1300C CONTINUE
1310C D0 110 I=1,25
1320C D0 105 J=1,5
1330C PRE1(J,1)=" "

```

Figure VI-11 Program DIALOGUE (cont'd)

```

1340      DO 110 K=1,3
1350      PRE2(K,I)=""
1360      PRE4(K,I)=0.0
1370C      BEGIN DIALOGUE WITH OPERATOR.
1380      CALL OUT(7)
1390      IF(YESNO(I)) 500,220,120
1400      CALL WHERE("3-1",120) IF(.NOT.SKIPI) GO TO 130
1410      CALL OUT(8)
1420      IF(YESNO(I)) 500,130,500
1430      CALL OUT(9)
1440      CALL WHERE("2-2",200) NR=1; NP=1; I=1; FLAG=.TRUE.; RAG=.FALSE.
1450      CALL PLUCK(TELETYPE,ARRAY,L,2,NTR,NTP)
1460      IF(ARRAY(1).EQ."END") GO TO 205
1470      DO 202 J=1,5
1480      W(J)=ARRAY(J)
1490      CALL STEM(W,S,L)
1500      IF(L.EQ.0) GO TO 201
1510      203 IF (PRE1(1,I).EQ."") GO TO 9203
1520      IF (1.6E-25) GO TO 205
1530      I=I+1; GO TO 203
1540      9203 DO 204 J=1,5
1545      IF (J.LE.3) PRE2(J,I)=S(J)
1550      204 PRE1(J,I)=W(J)
1560      BAG=LOOKUP(I)
1570      FLAG=FLAG.AND.BAG
1580      RAG=RAG.OR.BAG
1590      GO TO 201
1600      205 IF (FLAG) GO TO 300
1610      CALL OUT(10)
1620C      GO TO 300 IF ALL STEMS IN DICTIONARY. PRINT EXCEPTIONS IF NOT.
1630      DO 207 I=1,25
1640      IF(PRE3(1,I).NE.0) GO TO 207
1650      IF(PRE1(1,I).NE."") WRITE(TELETYPE,9900) (PRE1
1655      (K,I), K=1,5)
1660      DO 206 J=1,5

```

Figure VI-11 Program DIALOGUE (cont'd)

```

1670      IF(J.LE.3) PRE2(J,I)=-
1680 206  PRE1(J,I)=-
1690 207  CONTINUE
1700C    GO TO 300 UNLESS TABLE TOTALLY EMPTY
1710      IF (RAG) GO TO 300
1720      CALL OUT(11)
1730 210  CALL WHERE("5-2",210), CALL OUT(12)
1740C    CHANGE MODE?
1750      IF (YESN(1)) 500,211,120
1760 211  CALL OUT(13)
1770C    DOES USER WANT TO QUIT?
1780      IF(YESN(1)) 500,220,230
1790 220  CALL WHERE("1-2",220), NEWQ=.TRUE., GO TO 1310
1800 230  CALL OUT(26), STOP
1810 240  CALL WHERE("6-2",240), IF(RFL6) GO TO 500
1820      GO TO 210
1830C    QUERY IN-- FORM QUERY VECTOR--FIFTY LARGEST COMPONENTS
1840 300  CALL WHERE("4-3",300), L=1
1850      DO 309 I=1,25
1860      DO 309 J=1,3
1870      IF(PRE3(J,I).EQ.0) GO TO 309
1880      DO 301 K=1,50
1890      IF(PRE3(J,I).NE.QUERY1(K)) GO TO 301
1900      QUERY2(K)=QUERY2(K)+PRE4(J,I), GO TO 309
1910 301  CONTINUE
1920      IF (L.GT.50) GO TO 303
1930      QUERY1(L)=PRE3(J,I), QUERY2(L)=PRE4(J,I), L=L+1, GO TO 309
1940 303  SMALL=1E6
1950      DO 304 K=1,50
1960      IF (QUERY2(K).GE.SMALL) GO TO 304
1970      SMALL=QUERY2(K), LS=K
1980C
1990 304  CONTINUE
2000      IF(PRE4(J,I).LE.SMALL) GO TO 309
2010      QUERY1(LS)=PRE3(J,I), QUERY2(LS)=PRE4(J,I)

```

Figure VI-11 Program DIALOGUE (cont'd)

```

2020 309 CONTINUE
2040 314 CALL WHERE("5-3",314); IF(.NOT.PRINTQ) GO TO 330
2050C IF PRINTQ SET, USER CAN PRINT PRE & QUERY
2060 CALL OUT(15)
2070 IF(YESN(1)) 500,330,315
2080 315 CALL OUT(16)
-2085 NFULL=0
2090 DO 319 I=1,25
2100 IF(PRE3(1,I).GT.0) WRITE(TELETYPE,9901) (PRE1(J,I),J=1,5), (
21104 PRE2(K,I),K=1,3), (PRE3(L,I),PRE4(L,I),L=1,3)
2115 NFULL=NFULL+PRE3(1,I)
2120 319 CONTINUE
2125 IF(NFULL.EQ.0) CALL OUT(75)
2130 WRITE(TELETYPE, 9902)
2140 CALL OUT(17)
2150 IF(YESN(1)) 500,330,320
2160 320 CALL OUT(18)
2161 DO 323 K=1,46.5
2162 I=K+4
2163 DO 321 J=K,I
2164 IF(QUERY1(J).NE.0) GO TO 322
2165 321 CONTINUE
2166 GO TO 323
2167 322 WRITE(TELETYPE,9903) (QUERY1(11), QUERY2(11), 11=K,I)
2170 323 CONTINUE
2175 WRITE(TELETYPE,9902)
2180C NOW CORRELATE. FIRST CLEAR RETRIEVED LAST FLAG, FORM TERM
2190 330 NJX=0; SIG=0.0; FLAG=.FALSE.
2200 DO 331 I=1,50
2210 IF (QUERY2(I).LE.0) GO TO 331
2220 SIG=SIG+(QUERY2(I))**2
2230 331 TEMP6(I)=.FALSE.
2240C NOW CORRELATE QUERY AGAINST DOCUMENTS.
2250 COR=0; SIGD=0; IOLD=1
2260 332 READ("CONCEPTS",9904,END=342) I, (ICON(K), WT(K), K=1,5)

```

Figure VI-11 Program DIALOGUE (cont'd)

```

2270C      CHECK IF DOCUMENT EXCLUDED.
2280      DO 333 K=1,100
2290      IF (1.EQ.NONB(K)) GO TO 332
2300      CONTINUE
2310 333  IF (10LD.NE.1) GO TO 336
2320      DO 335 J=1,5
2330      IF(WT(J).GT.0) SIGD=SIGD+WT(J)**2
2340      DO 335 K=1,50
2350      IF (QUERY1(K).NE.ICON(J)) GO TO 335
2360      COR=COR+QUERY2(K)*WT(J)
2370 335  CONTINUE
2380      GO TO 332
2390 336  IF((SIGD*SIGD).LE.0) GO TO 341
2395      COR=COR/SQRT(SIGD*SIGD)
2400C      CORRELATION NOW FOUND. IF ZERO SKIP. IF NOT, CHECK IF ALREADY
2410C      IN TABLE. IF SO, MAKE SIGN OF ACCESSION NEGATIVE AS A 'RETR-
2420C      IEVED TWICE' FLAG AND MODIFY TABLE ENTRY. IF NOT, ENTER IF
2430C      ROOM EXISTS OR IT CORRELATES HIGHER THAN ANOTHER NEW HIT.
2440      IF(COR.LE.0) GO TO 341
2450      DO 337 J=1,50
2460      IF(TEMP1(J).EQ.10LD) GO TO 338
2470 337  CONTINUE
2480      GO TO 339
2490 338  TEMP1(J)=-TEMP1(J); TEMP3(J)=COR; TEMP6(J)=.TRUE.; GO TO 341
2500 339  SMALL=1E6; L1=0; L2=0
2510      DO 340 J=1, 50
2520      IF(TEMP1(J).EQ.0) L1=J
2530      IF(TEMP1(J).LT.0 .OR. (.NOT.TEMP6(J)) .OR.
2540      TEMP3(J).GE.SMALL) GO TO 340
2550      SMALL=TEMP3(J); L2=J
2560 340  CONTINUE
2570C      HAVE FOUND A PLACE IN TEMP TABLES OR L1=L2=0
2580      IF(L1.EQ.0 .AND. COR.GT.SMALL) L1=L2
2590      IF(L1.EQ.0) GO TO 341
2600      TEMP1(L1)=10LD

```

Figure VI-11 Program DIALOGUE (cont'd)

```

2610 TEMP2(L1)=IX; IX=IX+1; JX=JX+1; NJX=NJX+1
2620 TEMP3(L1)=C0R
2630 TEMP5(L1)=.FALSE.
2640 TEMP6(L1)=.TRUE.
2650 341 C0R=0; SIGD=0; I0LD=1; G0 T0 334
2660C NOWPERFORM RANKING ON NEW HITS AND REVERSE ANY ACC. N0. <0.
2670 342 D0 343 J=1,50
2680 IF(TEMP1(J).LT.0) TEMP1(J)=-TEMP1(J)
2690 343 IF(TEMP6(J)) TEMP4(J)=0
2700 D0 345 J=1,50
2710 BIG=0.0; L=0
2720 D0 344 K=1,50
2730 IF(TEMP4(K).NE.0 .OR. (.N0T.TEMP6(K)) .OR.
2740 TEMP3(K).LT.BIG ) G0 T0 344
2750 BIG=TEMP3(K); L=K
2760 344 CONTINUE
2770 IF(L.NE.0) TEMP4(L)=J
2780 345 CONTINUE
2785 BEGIN FILE "CONCEPTS"
2790C TEMP FILE NOW BUILT. TELL USER HOW MANY HITS FOR LAST QUERY.
2800 IF (NJX.LT.50) G0 T0 346
2810 CALL 0UT(19); G0 T0 350
2820 346 CALL 0UT(20); WRITE(TELETYPE,9905) NJX
2830 350 IF(NJX.EQ.0) G0 T0 240
2840C RETRIEVAL HAS BEEN PERFORMED. IF USER HAS SELECTED MODE #4
2850C HE MAY NOW PRINT TEMPORARY FILE.
2860 400 CALL WHERE("8-4",400); IF(.N0T.PRINTR) G0 T0 440
2870 420 CALL WHERE("9-4",420); CALL 0UT(21)
2880 IF(YESH0(1)) 500,440,430
2890 430 CALL 0UT(22)
2900 D0 435 I=1,50
2910 IF(TEMP1(I).EQ.0) G0 T0 435
2920 WRITE(TELETYPE,9906) TEMP1(I), TEMP2(I), TEMP3(I), TEMP4(I),
2930 TEMP5(I), TEMP6(I)
2940 435 CONTINUE

```

Figure VI-11 Program DIALOGUE (cont'd)

```

2945      WRITE(TELETYPE,9902)
2950      CALL OUT(23)
2960      IF (YESN(I)) 500,500,445
2970C     PRINT BIBLIOGRAPHIC DATA, IF NOT PRINTED BEFORE.
2980C
2990      J=1
3000      K=1
3010      IF(TEMP1(J).EQ.0) GO TO 475
3015      K=K+1
3020      WRITE(TELETYPE,9907) TEMP1(J), TEMP2(J)
3030      IF(TEMP5(J)) GO TO 460
3040      FLAG=D0CK(1,TEMP1(J), ARRAY, L)
3050      WRITE(TELETYPE,9913) (ARRAY(N),N=1,L)
3060      IF(FLAG) GO TO 456
3070      FLAG=D0CK(2,TEMP1(J), ARRAY, L)
3080      WRITE(TELETYPE,9913) (ARRAY(N),N=1,L)
3090      IF(FLAG) GO TO 457
3100      GO TO 470
3110      CALL OUT(24)
3120      TEMP5(J)=.TRUE.
3130      J=J+1
3140      IF(J.GT.50) GO TO 490
3160      IF(K.LE.5) GO TO 455
3170C     MORE DESIRED?
3175      WRITE(TELETYPE,9902)
3180      CALL OUT(3)
3190      IF(YESN(I)) 500,500,450
3200      WRITE(TELETYPE,9902); CALL OUT(25)
3210C     THIS IS THE OPTIONS SECTION.
3220      CALL WHERE("OPT",500); CALL OUT(27)
3230      NR=J; NP=1
3240      CALL PLUCK(TELETYPE,ARRAY,L,3,NTR,NTP)
3250      IF(ARRAY(1).EQ."HELP") GO TO 520
3260      DO 501 J=1,13
3270      IF (ARRAY(1).EQ.OPTIONS(J)) GO TO 580

```

Figure VI-11 Program DIALOGUE (cont'd)

```

3280 501 CONTINUE
3290 510 CALL OUT(31); GO TO 500
3300 520 CALL OUT(28)
3310 IF (OPTION) GO TO 540
3320 530 CALL OUT(30); IF(YESN(1)) 500,500,540
3330 540 CALL OUT(29); GO TO 500
3340 580 DEFLG=.FALSE.; RFLG=.TRUE.
3350 590 CALL WHERE("INDX",J); GO TO (1300,760,1400,600,1100,1130,
3360 1200,600,1330,1000,740,930,730), J
3370C QUERY MODIFICATION--FIRST CHECK TO SEE IF DOING DEC.-DEC.
3380 600 CALL WHERE("M0D",600); WFLG=.FALSE.
3390 IF (D0CD0C) GO TO 605
3400 CALL OUT(51); GO TO 610
3410 605 CALL OUT(59)
3420 610 IF (YESN(1)) 500,700,620
3430C ENTRY POINT FOR DOCUMENT-DOCUMENT CORRELATION.
3440 620 CALL WHERE("DDC",620)
3450 D0 622 I=1,25
3460 D0 621 J=1,3
3470 PRE2(J,I)=" "
3480 PRE3(J,I)=0
3490 621 PRE4(J,I)=0.0
3500 D0 622 J=1,5
3510 622 PRE1(J,I)=" "
3520 IF(.NOT.D0CD0C) GO TO 640
3530 630 CALL OUT(59)
3540 IF(YESN(1)) 500,640,650
3550 640 D0 641 I=1,50
3560 QUERY1(I)=0
3570 641 QUERY2(I)=0.0
3580 650 CALL OUT(52)
3590 F1=0; D0CD0C=.TRUE.
3600 660 CALL NUMBER(F1,ARG,NTR,NTP)
3610 IF(F1.EQ.0) GO TO 900
3620C NOW DEEP INTO DOCUMENT-DOCUMENT CORRELATION.

```

Figure VI-11 Program DIALOGUE (cont'd)

```

3630 661 READ("CONCEPTS",9904,END=664) I, (ICON(K), WT(K), K=1,5)
3640 IF(I.LT.ARG) GO TO 661
3650 IF(I.GT.ARG) GO TO 664
3660 DO 663 K=1,5
3670 L=0
3680 DO 662 J=1,50
3690 IF (QUERY1(J).EQ.0) L=J
3700 IF (ICON(K).EQ.0) GO TO 664
3710 IF (QUERY1(J).NE.ICON(K)) GO TO 662
3720 QUERY2(J)=QUERY2(J)+WT(K)
3730 GO TO 663
3740 662 CONTINUE
3750 IF (L.EQ.0) GO TO 663
3760 QUERY1(L)=ICON(K); QUERY2(L)=WT(K)
3770 663 CONTINUE
3775 GO TO 661
3780 664 BEGIN FILE "CONCEPTS"; GO TO 660
3790C ASK IF PRESENT QUERY TO BE CLEARED.
3800 700 CALL WHERE("10-7",700); CALL OUT(70)
3810 IF(YESNO(1)) 500,710,800
3820 710 CALL OUT(53)
3830 IF(YESNO(1)) 500,720,750
3840 720 IF(.NOT.WFLG) GO TO 900
3855 DO 9721 J=1,50
3856 QUERY1(J)=0
3857 9721 QUERY2(J)=0.0
3858 DO 729 I=1,25
3860 IF (PRE2(1,I).EQ." ") GO TO 729
3870 IF (.NOT.LOOKUP(1)) GO TO 729
3880 DO 729 J=1,3
3890 L=0
3900 IF(PRE3(J,I).EQ.0) GO TO 729
3910 DO 721 K=1,50
3920 IF(QUERY1(K).EQ.0) L=K
3930 IF(QUERY1(K).NE.PRE3(J,I)) GO TO 721

```

Figure VI-II Program DIALOGUE (cont'd)

```

3940 QUERY2(K)=QUERY2(K)+PRE4(J,I); GO TO 729
3950 CONTINUE
3960 IF(L.EQ.0) GO TO 729
3970 QUERY1(L)=PRE3(J,I); QUERY2(L)=PRE4(J,I)
3980 CONTINUE
3990C
4000 GO TO 900
4010C HERE IF PRESENT QUERY TO BE PRINTED.
4020 730 CALL WHERE("WRD",730); WRDFLG=.TRUE.; GO TO 750
4030 740 CALL WHERE("SEE",740); SEFLG=.TRUE.
4040 750 CALL OUT(54); L=0
4050 DO 751 I=1,25
4060 IF(PRE1(I,I).NE." ") WRITE(TELETYPE,9900) (PRE1(J,I),J=1,3)
4070 751 CONTINUE
4080 IF (SEFLG) GO TO 930
4090 GO TO 840
4100C HERE TO CLEAR EVERYTHING EXCEPT FILE OF RETRIEVED DOCUMENTS.
4110 760 CALL WHERE("CLR",760); CALL OUT(72)
4120 DO 761 I=1,50
4130 QUERY1(I)=0
4140 761 QUERY2(I)=0.0
4150 DO 763 I=1,25
4160 DO 762 J=1,5
4170 762 PRE1(J,I)=" "
4180 DO 763 K=1,3
4190 PRE3(K,I)=0; PRE4(K,I)=0.0
4200 763 PRE2(K,I)=" "
4210 DBCDC=.FALSE.; GO TO 500
4220C NOW REPLACE OLD QUERY. FIRST CLEAR PRESENT.
4230 800 CALL WHERE("12-8",F00)
4240 DO 802 I=1,25
4250 DO 801 J=1,5
4260 801 PRE1(J,I)=" "
4270 DO 802 K=1,3
4280 802 PRE2(K,I)=" "; PRE3(K,I)=0

```

Figure VI-11 Program DIALOGUE (cont'd)

```

4290 802   PRE4(K,I)=0
4300 810   DECD0C=.FALSE.; WFLG=.TRUE.
4310       CALL 0UT(56); I=1
4320 815   NR=1; NP=1
4330 816   CALL PLUCK(TELETYPE,ARRAY,L,2,NTR,NTP;
4340       IF(L.EQ.0 .OR. ARRAY(1).EQ."END") GO TO 830
4350       DO 817 J=1,5
4360 817   W(J)=ARRAY(J)
4370       CALL STEM(W,S,L)
4380       IF (L.EQ.0) GO TO 816
4390       I=1
4400 818   IF(PRE1(1,1).EQ." ") GO TO 819
4410       I=I+1; IF(I.GT.25) GO TO 830
4420       GO TO 818
4430 819   DO 820 J=1,5
4440 820   PRE1(J,1)=W(J)
4450       DO 821 J=1,3
4460 821   PRE2(J,1)=S(J)
4470       GO TO 816
4480C      NOW ALL NEW, NON-COMMON QUERY WORDS AND THEIR STEMS ARE IN PRE.
4490 830   FLAG=.TRUE.
4500       DO 832 I=1,25
4505       IF(PRE1(1,1).EQ." ") GO TO 832
4510       IF(L00KUP(I)) GO TO 832
4520       IF(FLAG) CALL 0UT(10)
4530       FLAG=.FALSE.; WRITE(TELETYPE,9900) (PRE1(K,1),K=1,5)
4540       DO 831 K=1,5
4550       IF(K.LE.3) PRE2(K,1)=" "
4560 831   PRE1(K,1)=" "
4570 832   CONTINUE
4600       GO TO 850
4610C      ASK IF ANY WORDS TO BE ADDED.
4620 840   CALL WHERE("13-8",840); CALL 0UT(55)
4630       IF(YESN0(I)) 500,850,810
4640C      ASK IF ANY WORDS TO BE DELETED

```

Figure VI-11 Program DIALOGUE (cont'd)

```

4650 850 CALL OUT(57)
4660 IF(YESN(I)) 500,700,860
4670 860 D0CD0C=.FALSE.;WFLG=.TRUE.;NR=1;NP=1;FLAG=.TRUE.
4675 CALL OUT(56)
4680 861 CALL PLUCK(TELETYPE,ARRAY,L,2,NTR,NTP)
4685 IF (L.GT.0) M=L/4 + MINO(1, (L-4*(L/4)))
4690 IF(L.EQ.0 .OR. ARRAY(1).EQ."END") G0 T0 870
4700 D0 865 I=1,25
4710 D0 862 J=1,5
4720 IF(ARRAY(J).NE.PRE1(J,I)) G0 T0 865
4730 862 CONTINUE
4740 D0 863 K=1,5
4750 863 PRE1(K,I)=" "
4760 D0 864 L=1,3
4770 PRE2(L,I)=" "
4780 PRE3(L,I)=0
4790 864 PRE4(L,I)=0.0
4800 G0 T0 861
4810 865 CONTINUE
4820 IF (FLAG) CALL OUT(58)
4830 WRITE(TELETYPE,9900) (ARRAY(K),K=1,M)
4840 FLAG=.FALSE.; G0 T0 861
4850 870 IF(.NOT.FLAG) G0 T0 850
4860 IF(.NOT.WRDFLG) G0 T0 700
4870 WRDFLG=.FALSE.; G0 T0 500
4880C HERE WHEN QUERY VECTOR T0 BE INSPECTED/MODIFIED--PERHAPS.
4890 900 CALL WHERE("11-9",900); CALL OUT(60)
4900 IF(YESN(I)) 500,910,935
4910 910 CALL WHERE("14-9",910); CALL OUT(65)
4920 IF(YESN(I)) 500,600,920
4930 920 D0 921 I=1,50
4940 IF(QUER/1(I).NE.0) G0 T0 1000
4950 921 CONTINUE
4960 CALL OUT(66)
4970C QUERY VECTOR IS NULL!

```

Figure VI-11 Program DIALOGUE (cont'd)

```

4980 IF (YESN0(I)) 500,500,1130
4990C HERE QUERY VECT0K T0 BE INSPECTED/M0DIFIED--F0K SURE
5000 930 CALL WHERE("WGT",930); CALL 0UT(71)
5010C
5020 935 CALL 0UT(18)
5030 D0 938 I=1,46,5
5040 L=I+4
5050 D0 936 K=1,L
5060 IF(QUERY1(K).NE.0) G0 T0 937
5070 936 C0NTINUE
5080 G0 T0 938
5090 937 WRITE(TELETYPE,9903) (QUERY1(J),QUERY2(J),J=1,L)
5100 938 C0NTINUE
5105 WRITE(TELETYPE,9902)
5110 IF(.NOT.SEEFLG) G0 T0 940
5120 SEEFLG=.FALSE.; G0 T0 500
5130C VECT0K HAS BEEN PRINTED--IS IT T0 BE M0DIFIED.
5140 940 CALL 0UT(61)
5150 IF (YESN0(I)) 500,910,945
5160 945 CALL 0UT(62)
5170 950 READ: I,WX
5180 IF(I.GE.1 .AND. I.LE.1500) G0 T0 960
5190 CALL 0UT(64); G0 T0 970
5200 960 L=0
5210 D0 961 J=1,50
5220 IF(QUERY1(J).EQ.0) L=J
5230 IF(QUERY1(J).NE.1) G0 T0 961
5240 QUERY2(J)=QUERY2(J)+WX; G0 T0 970
5250 961 C0NTINUE
5260 IF(L.EQ.0) G0 T0 970
5270 QUERY1(L)=I; QUERY2(L)=WX
5280C CHECK IF ANY M0RE WANTED
5290 970 CALL 0UT(3)
5300 IF(YESN0(I)) 500,910,980
5310 980 CALL 0UT(63); G0 T0 950

```

Figure VI-11 Program DIALOGUE (cont'd)

```

5320C      ENTER HERE BEFORE ACTUAL RETRIEVAL.
5330 1000 CALL WHERE("RET",1000); IF(JX.EQ.0) CALL OUT(41)
5340      CALL OUT(43)
5350      IF(YESNO(I)) 500,1020,1010
5360 1010 L=100
5370      DO 1012 J=1,50
5380      DO 1011 I=1,100
5390      IF(NON0(101-I).EQ.0) L=101-I
5400      IF(TEMP(I,J).EQ.0 .OR. TEMP(I,J).EQ. NON0(I)) GO TO 1012
5410 1011 CONTINUE
5420      NON0(L)=TEMP(I,J)
5430 1012 CONTINUE
5440      GO TO 1030
5450 1020 DO 1021 I=1,100
5460 1021 NON0(I)=0
5470 1030 CALL OUT(44)
5480C      CHECK IF PRINT SUPPRESS TO BE REMOVED.
5490      IF(YESNO(I)) 500,1040,1050
5500 1040 DO 1041 I=1,50
5510 1041 TEMPS(I)=.FALSE.
5520C      DELETE ANY FROM TEMP FILE?
5530 1050 CALL WHERE("1510",1050); DEFLG=.FALSE.
5540      IF(JX.EQ.50) GO TO 1070
5550      IF(JX.EQ. 0) GO TO 1090
5560      WRITE(TELETYPE,9915) 50-JX
5570      CALL OUT(45)
5580      IF (YESNO(I)) 500,1090,1080
5590 1070 CALL OUT(46)
5600 1080 DEFLG=.TRUE.; GO TO 1100
5610 1090 CALL OUT(42)
5620      LMAX=50
5630      WRITE(TELETYPE,9908) IX
5640      GO TO 314
5650C      NOW DELETE DOCUMENTS FROM TEMP FILE.
5660 1100 CALL WHERE("DEL",1100); CALL OUT(36)

```

Figure VI-11 Program DIALOGUE (cont'd)

```

5670      IF(DEFLG) CALL OUT(37)
5680      CALL OUT(38); F1=0
5690 1101  CALL NUMBER(F1,ARG,NTR,NTP)
5700      IF (F1.EQ.0) GO TO 1120
5710      DO 1102 J=1,50
5720      IF(ARG.NE.TEMP1(J)) GO TO 1102
5730      TEMP1(J)=0;TEMP2(J)=0;TEMP3(J)=0.0;TEMP4(J)=0
5740      TEMP5(J)=.FALSE.; TEMP6(J)=.FALSE.
5750      JX=JX-1
5760 1102  CONTINUE
5770      GO TO 1101
5780 1120  IF (DEFLG) GO TO 1050
5790      GO TO 500
5800C      END OF QUERY SEQUENCE
5810 1130  CALL WHERE("END",1130)
5820      DO 1131 J=1,50
5830      TEMP1(J)=0;TEMP2(J)=0;TEMP3(J)=0.0
5840      TEMP4(J)=0;TEMP5(J)=.FALSE.;TEMP6(J)=.FALSE.
5850      QUERY1(J)=0;QUERY2(J)=0.0; NONG(J)=0
5860 1131  NONG(J+50)=0
5870      DO 1133 I=1,25
5880      DO 1132 J=1,5
5890      PRE1(J,I)=" "
5900      DO 1133 J=1,3
5910      PRE2(J,I)=" "
5920 1133  PRE4(J,I)=0.0
5930      IX=1;JX=0
5940      NEWQ=.FALSE.;DEFLG=.FALSE.;DQCDQC=.FALSE.;TERSE=.FALSE.
5950      SKIP1=.FALSE.;PRINTQ=.FALSE.;PRINTR=.FALSE.;OPTION=.FALSE.
5960      RFLG=.FALSE.; CALL OUT(32); GO TO 210
5970C      HERE TO PRINT DOCUMENT DATA.
5980 1200  CALL WHERE("DQC",1200); IF(JX.EQ.0) GO TO 1205
5990      CALL OUT(47)
6000      IF(YESNO(1)) 500,1205,430
6010 1205  F1=0; CALL OUT(68)

```

Figure VI-11 Program DIALOGUE (cont'd)

```

6020 1210 CALL NUMBER(F1,ARG,NTR,NTP); IF(F1.EQ.0) GO TO 500
6030C
6040 WRITE(TELETYPE,9909) ARG
6050 DO 1211 J=1,50
6060 IF(TEMP1(J).EQ.ARG) GO TO 1212
6070 1211 CONTINUE
6080 GO TO 1220
6090 1212 WRITE(TELETYPE,9910) TEMP2(J),TEMP3(J),TEMP4(J)
6100 IF(.NOT.TEMP6(J)) WRITE(TELETYPE,9911)
6110 WRITE(TELETYPE,9912)
6120 IF(.NOT.TEMP5(J)) GO TO 1220
6130 PS=.TRUE.; GO TO 1225
6140 PS=.FALSE.
6150 DO 1226 I=1,100
6160 IF(MON(I).NE.ARG) GO TO 1226
6170 CALL OUT(48); GO TO 1230
6180 1226 CONTINUE
6190 1230 IF (PS) GO TO 1235
6200 1231 FLAG=D0CK(1,ARG,ARRAY,L)
6210 WRITE(TELETYPE,9913) (ARRAY(K), K=1,L)
6220 IF (FLAG) GO TO 1231
6230 1232 FLAG=D0CK(2,ARG,ARRAY,L)
6240 WRITE(TELETYPE,9913) (ARRAY(K),K=1,L)
6250 IF (FLAG) GO TO 1232
6260 GO TO 1240
6270 1235 CALL OUT(49)
6280 IF(YESN0(K)) 500,1240,1231
6290 1240 CALL OUT(50)
6300 1245 IF(YESN0(K)) 500,1270,1260
6310 1260 FLAG=D0CK(3,ARG,ARRAY,L)
6320 WRITE(TELETYPE,9913) (ARRAY(K),K=1,L)
6330 IF (FLAG) GO TO 1260
6340 IF (F1.EQ.(-1)) GO TO 1210
6350 F1=4; GO TO 1280
6360 1270 IF (MOD(F1,5).NE.0) GO TO 1210

```

Figure VI-11 Program DIALOGUE (cont'd)

```

6370 1280 WRITE (TELETYPE,9902); CALL OUT(33)
6380 IF (YESN(K)) 500,1290,1210
6390 1290 F1=-1; G0 T0 1210
6400C THIS SECTION ALLOWS USER TO CHANGE MODES OF OPERATION.
6410 1300 CALL -WHERE("CHG",1300); NEWQ=.FALSE.
6420 1310 CALL WHERE("7-13",1310); TEKSE=.FALSE.; SKIP1=.FALSE.; PRINTQ=
6430 .FALSE.; PRINTR=.FALSE.; OPTION=.FALSE.
6440 CALL OUT(34)
6450 IF(YESN(I)) 500,1316,1314
6460 1314 CALL OUT(35)
6470 1316 CALL OUT(14)
6480 NR=1; NP=1
6490 1317 CALL PLUCK(TELETYPE,ARRAY,L,3,NTR,NTP)
6500 D0 1318 I=1,6
6510 IF (ARRAY(I) .EQ. VALUES(I)) G0 T0 1319
6520 1318 CONTINUE
6530 CALL OUT(73); G0 T0 1300
6540C BRANCH ON OPTION SELECTED
6550 1319 G0 T0 (1320,1321,1322,1323,1324,1325), I
6560 1320 TEKSE=.TRUE.; G0 T0 1317
6570 1321 SKIP1=.TRUE.; G0 T0 1317
6580 1322 PRINTQ=.TRUE.; G0 T0 1317
6590 1323 PRINTR=.TRUE.; G0 T0 1317
6600 1324 OPTION=.TRUE.; G0 T0 1317
6610C NEW OPTIONS SELECTED
6620 1325 IF(NEWQ) G0 T0 120
6630 G0 T0 500
6640C THIS PART CALLED TO BUILD A FILE FOR OFFLINE LISTING.
6650 1330 CALL WHERE("OFF",1330); F1=0
6660 CALL OUT(67)
6670 1340 CALL NUMBER(F1,ARG,NTR,NTP)
6680 IF (F1.EQ.0) G0 T0 500
6690 WRITE("OFFLINE",9909, ARG
6700 D0 1350 J=1,3
6710 1341 FLAG=D0CK(J,ARG,ARRAY,L)

```

Figure VI-11 Program DIALOGUE (cont'd)

```

6720 WRITE("OFFLINE",9913) (ARRAY(K),K=1,L)
6730 IF(FLAG) GO TO 1341
6740 1350 CONTINUE
6750 CALL OUT(40)
6760 GO TO 1340
6770C THIS PART PRINTS DOCUMENTS' CONCEPT-WEIGHT DATA.
6780 1400 CALL WHERE("CON",1400), FI=Q
6800 1410 CALL NUMBER(FI,ARG,NTR,NTP)
6810 IF (FI.EQ.0) GO TO 500
6820 CALL OUT(39)
6830 WRITE(TELETYPE,9914) ARG, CALL OUT(18)
6840 1420 READ("CONCEPTS",9904,END=1440) I, (ICON(K), WT(K), K=1,5)
6850 IF (I.GT.ARG) GO TO 1440
6860 IF (I.LT.ARG) GO TO 1420
6870 WRITE(TELETYPE,9903) (ICON(K),WT(K), K=1,5)
6880 GO TO 1420
6890 1440 BEGIN FILE "CONCEPTS"
6895 WRITE(TELETYPE,9902)
6900 IF(MID(FI,5).NE.0) GO TO 1410
6910 CALL OUT(33)
6920 IF ('ESN0(I)) 500,500,1410
6930 9900 FORMAT(1X,5A4)
6940 9901 FORMAT(1X,5A4,1X,3A4,1X,3(15,F9.5))
6950 9902 FORMAT(2X)
6960 9903 FORMAT(5(14,F9.5))
6970 9904 FORMAT(16,5(14,F9.5))
6980 9905 FORMAT(2H4,14,1H.)
6990 9906 FORMAT(16,113,E18.4,19,L11,"/",L1)
7000 9907 FORMAT("// " ACC. N0,"16,"/" TEMP. ID.",17,".")
7010 9908 FORMAT(1H4,13,1H.)
7020 9909 FORMAT("/" ACC. N0,"16,1H.)
7030 9910 FORMAT("4 ID.",13," CORR.",F6.3," RANK",13,".")
7040 9911 FORMAT("4N0T ")
7050C

```

Figure VI-11 Program DIALOGUE (cont'd)

```
7060 9912 FORMAT("RETRIEVED BY LAST QUERY")
7070 9913 FORMAT(1X,18A4)
7080 9914 FORMAT("4", 16, ".")
7085 9915 FORMAT(" THERE ARE",13," SPACES IN THE TEMPORARY FILE ")
7090      END
```

Figure VI-11 Program DIALOGUE (concluded)

```

1000C CONGRA 12/30/69
1010 ASCII D(900,3), X(18), W(5), S(5), FLAG
1020 INTEGER COND(900), CON(50)
1030 DIMENSION WTD(900), WT(50)
1040 DATA ID, NR, NP/3*1/
1050 DO 100 I=1,900
1060 100 READ ("DICTNRY",101) (D(I,L),L=1,3), COND(I)
1070 101 FORMAT (3A4,1X,14)
1080 105 DO 110 K=1,900
1090 110 WTD(K)=0.0
1100 DO 115 K=1,50
1110 CON(K)=0
1120 115 WT(K)=0
1130 WTMAX=0.0
1140 PRINT:"PROCESS NO. ", ID
1150 READ: FLAG
1160 IF (FLAG.NE."N0") GO TO 120
1170 END FILE "CONCEPTS"
1180 STOP
1190 120 CALL PLUCK("DATA1",X,L,2,NR,NP)
2000 IF (X(1).EQ."**EN") GO TO 200
2010 DO 121 K=1,5
2020 121 W(K)=X(K)
2030 CALL STEM(W,S,L)
2040 IF (L.EQ.0) GO TO 120
2050 DO 130 K=1,900
2060 DO 125 L=1,3
2070 IF (S(L).NE.D(K,L))GO TO 130
2080 125 CONTINUE
2090 GO TO 140
2100 130 CONTINUE
2110 GO TO 120
2120 140 WTD(K)=WTD(K)+1.0
2130 WTMAX=AMAX1(WTMAX,WTD(K))
2140 GO TO 120

```

Figure VI-12 Program CONGRA (cont'd)

```

2150 200 LIMIT=1
2160 DO 210 K=1,900
2170 IF (WTD(K).EQ.0) GO TO 210
2180 CON(LIMIT)=CON(K)
2190 WT(LIMIT)=WTD(K)/WTMAX
2200 LIMIT=LIMIT+1
2210 IF(LIMIT.GT.50) GO TO 220
2220 210 CONTINUE
2230 LIMIT=LIMIT-1
2240 PRINT: LIMIT, "CONCEPTS AT LINE NO.", NR, "."
2250 K=1
2260 225 L=K+4
2270 WRITE("CONCEPTS",226) ID, (CON(J),WT(J), J=K,L)
2280 226 FORMAT (I6, 5(I4,F9.5))
2290 IF (L.GE.LIMIT) GO TO 230
2300 K=L+1
2310 GO TO 225
2320 230 ID=ID+1
2330 GO TO 105
2340 END

```

Figure VI-12 Program CONGRA (concluded)

```

100C      PROGRAM TO GENERATE TEST DICTIONARY.
110      ASCII X(18), DICT(5000,3), S(5)
120      LOGICAL FLAG
130      DATA NR,NP,DICT/2*1,15000* " "
140      PRINT: "MAXIMUM NUMBER OF ENTRIES"
145      IKYP00=0
150      READ: IMAX
160      I=0
170      CALL PLUCK("DATA1", X, L, 2, NR, NP)
180      IF(NR.EQ.0) GO TO 15
190      IF(X(1).NE."**AB") GO TO 3
191      IKYP00=IKYP00+1
192      PRINT: IKYP00, I, " MORE?"
193      READ: S(1)
194      IF(S(1).EQ."N0") GO TO 15
200      FLAG=.TRUE.; GO TO 2
210      IF(X(1).NE."**AU" .AND. X(1).NE."**TI" .AND. X(1).NE."**EN"
220      .AND. X(1).NE."**C0") GO TO 31
230      FLAG=.FALSE.; GO TO 2
240      IF (FLAG) GO TO 4
250      NR=NR+1; NP=1; GO TO 2
260      IF(X(1).LT."A " .OR. X(1).GT."ZZZZ") GO TO 2
270      CALL STEM(X,S,L)
280      IF(L.EQ.0) GO TO 2
290      J=1
300      IF(I.EQ.0) GO TO 13
310      IF(S(1).LT.DICT(J,1)) GO TO 10
320      IF(S(1).GT.DICT(J,1)) GO TO 9
330      IF(S(2).LT.DICT(J,2)) GO TO 10
340      IF(S(2).GT.DICT(J,2)) GO TO 9
350      IF(S(3).LT.DICT(J,3)) GO TO 10
360      IF(S(3).GT.DICT(J,3)) GO TO 9
365      GO TO 2
370      J=J+1

```

Figure VI-13 Program DICGEN (cont'd)

```

380 IF(J.GT.1) GO TO 13
390 GO TO 6
400 K=I
410 10 KK=K+1
420 DO 99 KKK=1,3
430 99 DICT(KK,KKK)=DICT(K,KKK)
440 IF(K.EQ.J) GO TO 13
450 12 K=K-1; GO TO 11
460 13 DO 97 KKK=1,3
470 97 DICT(J,KKK)=S(KKK)
475 PRINT 69
476 69 FORMAT(2H&)
480 IF(I.GE.IMAX) GO TO 15
490 14 I=I+1; GO TO 2
500 15 PRINT 15, I
510 16 FORMAT("THERE ARE", I5, " STEMS." ///)
520 DO 89 J=1,I
530 89 WRITE("DICTNRY",17) (DICT(J,K),K=1,3),J,1.0,0.0,0.0,0.0
535 17 FORMAT(3A4,1X,3(I4,F9.5))
537 END FILE "DICTNRY"
540 STOP; END

```

Figure VI-13 Program DICGEN (concluded)

## SECTION VII

### REFERENCES

1. Baker, A. W., Hoffmann, J. P. and Smith, J. L. Colex User's Manual. System Development Corporation, Technical Memorandum TM-DA-(L)-15/001/00, undated.
2. Bleier, Robert E., Treating hierarchical data structures in the SDC time-shared data management system (TDMS), Proceedings of 22nd National Conference, Association for Computing Machinery. Thompson Book Company, Washington, 1967, p. 41.
3. Giering, R. H., Information Processing and the Data Spectrum. Data Corporation publication, October 1967.
4. Giering, R. H., Analysis of Existing and Proposed Data Handling Systems. Data Corporation Technical Note DTN-69-9, October 1967.
5. Lowe, T. C. and Roberts, D. C., On-Line Retrieval Interim Report. Informatics Inc., Bethesda, Md.. Report No. TR-69-1090-1A. August 1969, RADC Report No. RADC-TR-69-304.
6. Lowe, Thomas C., Design Principles for an On-Line Information Retrieval System. Ph.D. Dissertation, U. of Pennsylvania, Philadelphia, 1966.
7. Nance, J. W. and Lathrop, J. W., System Design Specification General Purpose Orbit. SDC Technical Memorandum TM-DA-20/000/00, 1968.
8. Stevens, Mary Elizabeth, Automatic Indexing: A State of the Art Report. National Bureau of Standards Monograph 71, 1965.
9. Welch, Noreen O., A Survey of Five On-Line Retrieval Systems. Mitre Corporation Report MTP-322, August 1968, (in support of COSATI Panel 2).

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D		
(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)		
1. ORIGINATING ACTIVITY (Corporate author) Informatics, Inc. 4720 Montgomery Lane Bethesda, Maryland 20014		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED
		2b. GROUP N/A
3. REPORT TITLE ON-LINE RETRIEVAL		
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Final Report 1 August 1969 - 23 February 1970		
5. AUTHOR(S) (First name, middle initial, last name) Thomas C. Lowe David C. Roberts		
6. REPORT DATE April 1970	7a. TOTAL NO. OF PAGES 198	7b. NO. OF REFS 9
8a. CONTRACT OR GRANT NO. F30602-69-C-0038	8a. ORIGINATOR'S REPORT NUMBER(S) TR-69-1090-2	
8b. PROJECT NO. 4594		
c. Task 459401	8b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC-TR-70-44	
d.		
10. DISTRIBUTION STATEMENT Distribution of this document is unlimited. It may be released to the Clearinghouse, Department of Commerce, for sale to the general public.		
11. SUPPLEMENTARY NOTES Report on an on-line information storage and retrieval system		12. SPONSORING MILITARY ACTIVITY Rome Air Development Center (EMIDB) Griffiss Air Force Base, New York 13440
13. ABSTRACT <p>This report is concerned with the implementation of an on-line information storage and retrieval system for the Rome Air Development Center. This system is to incorporate techniques of automatic document classification for a large document collection in an interactive environment. Following a review of the system design, the implementation of the system executive is described in detail. Because this executive program also governs communications between the user and the system, it must be a communications package, a training aid, a file building program, and an executive program all in one.</p>		

DD FORM 1 NOV 66 1473

UNCLASSIFIED

Security Classification

UNCLASSIFIED

Security Classification

14	KEY WORDS	LINK A		LINK B		LINK C	
		ROLE	WT	ROLE	WT	ROLE	WT
	Information Retrieval Automatic Document Classification Content Analysis On-Line Systems Searching						

UNCLASSIFIED

Security Classification

RADC-TR-70-44

April 1970

ERRATA - May 1971

AD-706866  
Please attach the two pages to subject document, entitled, "On-Line  
Retrieval, RADC-TR-70-44 (AD#706866), Unclassified report, dated April 1970

Rome Air Development Center  
Air Force Systems Command  
Griffiss Air Force Base, New York

## ERRATA

The reader of this report should be aware of the contributions made by Dr. Gerard Salton of the Dept. of Computer Sciences at Cornell University through his research in information storage and retrieval, particularly the SMART document retrieval system. Many of the notions and methodologies concerned with document retrieval described in this report are attributable to him. For example, the concept vector technique as used in this system, and fundamental to it, should be credited to Dr. Salton as well as the techniques used for document-document and query-document correlation. Publications authored by Dr. Salton and his students (1A, 2A, 3A, 4A, 5A) were consulted and significantly influenced the overall system design. The reader is encouraged to refer to these documents as well as to RADC-TR-69-304 (6A) for further details.

## REFERENCES

- 1A Salton, G. and Lesk, M.E., The SMART Automatic Document Retrieval System - an illustration. C. ACM 8, 6(June 1965), pp. 391-398.
- 2A. Salton, Gerard, The Evaluation of Automatic Retrieval Procedures selected test results using the SMART System. American Documentation 16, 3(July 1965) pp. 209-222.
- 3A. Salton, G., et.al., The SMART System - Retrieval Results and Future Plans, in Rep No. ISR-11 to the National Science Foundation, Dept of Computer Science, Cornell University, Ithaca, NY, June 1966, I-4.
- 4A. Salton, G. and Lesk, M.E., Computer Evaluation of Indexing and Text Processing. J. ACM 15, 1(Jan 1967) 8-36.
- 5A. Salton, G., Automatic Information Organization and Retrieval. McGraw-Hill, New York, 1968.
- 6A. Lowe, T.C. and Roberts, D.C., On-Line Retrieval, Informatics Inc., Bethesda, MD. Report No. TR-69-1090-1A, August 1969, RADC Report No. RADC-TR-69-304.